Word Search: Puzzle Program Inside For Commodore, Atari, Apple, IBM, & TI

COMPUTE!

$2.95
September
1985
Issue 64
Vol. 7, No. 9
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

THE AMIGA FROM COMMODORE: An In-Depth Review

Programs Inside:

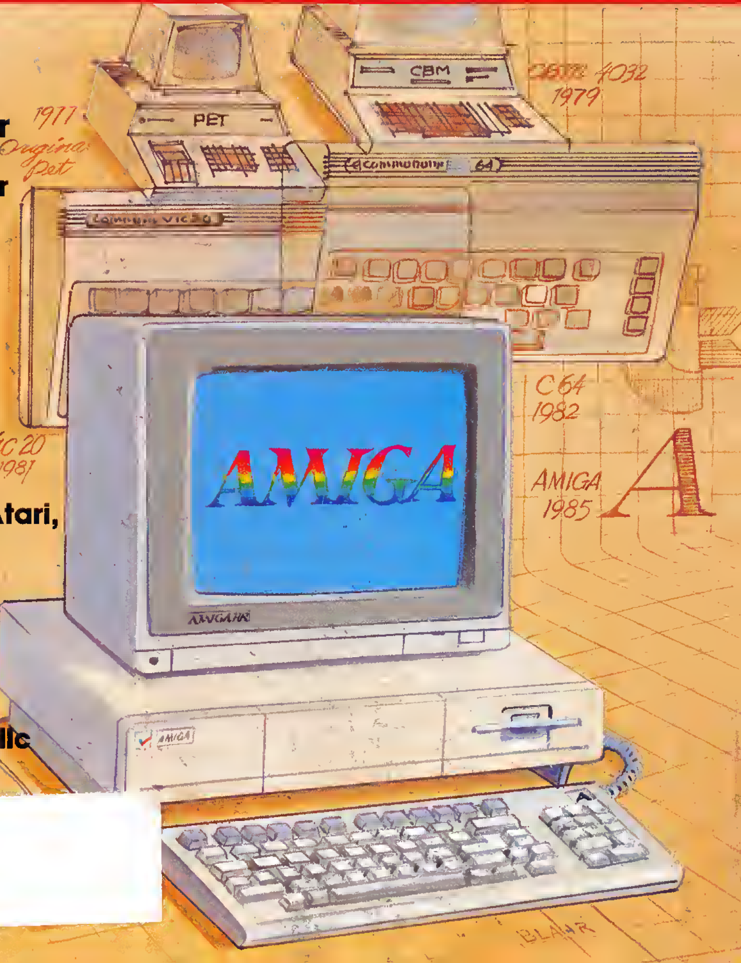
Disk Commander
Add 18 Powerful Commands To Your Commodore 64

Atari Animation

All About IBM Batch Files

Jump Search
For Commodore, Atari, Apple, IBM, And TI

Easy Apple Screen Editing
Enhanced BASIC Line Editor For Apple II+, IIe, IIc



With Commodore 128's instead of Apple IIc's, these kids would be on computers instead of in line.

Meet the Cammadare 128.™ The new personol computer that's already destined to be at the heod of its class. It nat anly autsmorts the Apple® IIc in price, it comes out woy oheod in performonce.

A lawer price is welcome news ta any tight school budget. But it's not the only woy the 128 soves you maney. You anly need to buy one external disk drive to netwark eight Commodore 128's fram a teacher's desk. The Apple IIc hos o built-in disk drive. Thot odds a built-in extro cost far a feature yaur clossraam daesn't reolly need. There's even more ta the Cammadare 128 thon being oble ta put more students on computers for less money.

There's more intelligence.

As your students grow smorter, sa daes the Cammodore 128. It's o computer they'll find more useful becouse it's copoble of exponding fram 128K to 512K memary. The Apple IIc daesn't expond, moking it less versatile. It also doesn't have the Cammodore 128's exponded keyboard that offers mare cammonds for easier pragromming ond more varied use of grophics ond text. Or o numeric keypad thot's a reol necessity when using o computer in moth ar science closses.

More software.

The Cammodare 128 is one new personol computer yau wan't hove to woit for software

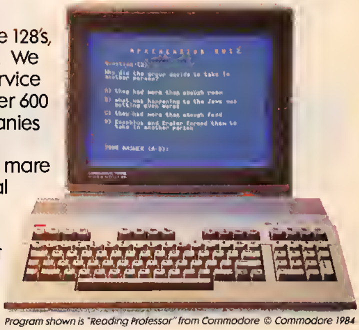
© Commodore 1985

to catch up to. It's compatible with aver 3,000 pragrams and peripherals designed far the Cammadare 64™, many af which yaur schaal may have. And in the 128 and CP/M® mades, there are pragrams far students wha want ta get dawn ta business and learn spreadsheets, file management ar ather prafessianal uses.

More convenience.

Here's a feature you'll alsa find intelligent. Shauld anything ga wrong with yaur Commodore 128's, just call tall-free 800-247-9000. We have an exclusive schaal service agreement with RCA and aver 600 independent service companies ta give you fast repair.

When it cames ta fitting mare computers inta a tight schaal budget, came size up the Cammadare 128. For mare intelligence, and a price that makes sense, you'll find it's ane sure way ta lawer the cast af a higher education.



```
        COMPREHENSION QUIZ
Question (2)
Why did the group decide to take in
another person?

A) they had more than enough room
B) what was happening to the Jews was
   getting even worse
C) they had more than enough food
D) Koophuis and Kraler forced them to
   take in another person


YOUR ANSWER (A-D):
```

Program shown is "Reading Professor" from Commodore © Commodore 1984

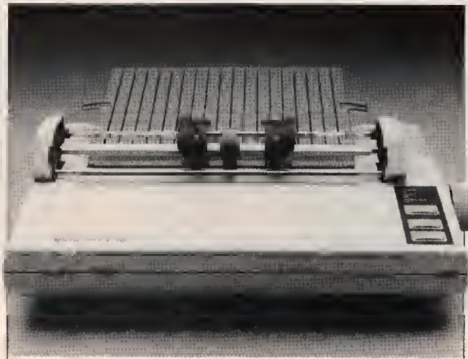
COMMODORE 128 PERSONAL COMPUTER

A Higher Intelligence

®Apple is a registered trademark of Apple Computer, Inc.
® CP/M is a registered trademark of Digital Research, Inc

Epson *all* your conc

You have your computer. Now you want a printer. And you have natural concerns. Can you get a serious printer without having to spend serious money? What about compatibility? How reliable are they? Read on.



YEARS AHEAD OF THE COMPETITION

While most other printers only offer standard 90-day warranties, we stand behind Epson printers for a full year.

Of course, the warranty is one benefit of owning an Epson that very few people ever have the opportunity to take advantage of.

Another benefit you probably won't use is the industry's strongest local service network. But it's nice to know it's there.

INCOMPATIBILITY: *the irreconcilable difference.*

Software drives your computer. Software also drives your printer. And it's very possible that your computer software will not be compatible with your printer.

This can be shocking when the p's on your monitor come out as q's on your printer.

But Epson printers are the standard to which virtually all major computer software is written. So whether you have an IBM® PC or the newest name on the market, there are Epson printers for virtually any hardware/software combination.

Is it an Epson ?????OR????? Isn't it an Epson?

For over a decade, major computer companies have staked their own reputations on the reliability and performance of Epson printers by coming to us for the printers they sell with *their* computer systems.

The most TRUSTED printers in the WORLD

Today, more homes and businesses rely on Epson printers than any other printers in the world. And the reason is this. Day to day, year to year, you can count on them. Epson printers are built for years of worry-free service. In fact, most of our printers have been working longer than most of our competitors have been in the business.

You know you need a printer to get the best from your computer. And, if you're going to spend your good money, you shouldn't have to settle for something unproven or less than reliable.

Check around. Now you can afford an Epson. Why settle for anything else?

answers erns in print.

HARD COPY DECISIONS MADE EASY.

For almost any computer. With the Epson Spectrum™ LX-80 printer, and interface cable, you can work with almost any computer. Most often used with the Apple IIe® and IBM PC, it prints out 100 characters per second in draft mode. And 16 characters per second, Near Letter Quality. With SelecType push-button typestyle control and excellent graphic capabilities, it offers a lot of performance for not a lot of money. Available through Epson authorized computer specialty stores.

"So the question is: 'Which Epson?'"

For the Apple IIc® and IBM® PCjr. The Epson Spectrum™ LX-90 is the perfect match for the Apple IIc and IBM PCjr. Also printing 100 characters per second, draft, and 16 characters per second in Near Letter Quality mode, the Spectrum LX-90 features friction and tractor feeds for both plain paper and continuous form computer paper. This makes it better for business applications. It uses a Printer Interface Cartridge (PIC™) to assure maximum compatibility with your computer.

ONLY EPSON PRINTERS COULD GENERATE PRINT LIKE THIS

❝Thank you for making such a rugged yet so fantastic piece of hardware.❞
L.F.G., Martinez, CA

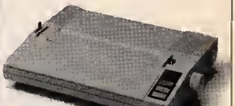
❝...a relief in this industry often faulted by products designed to last about ninety days.❞
John Cameron, Medical Director, Raleigh Hills Hospital

❝I have had my Epson about six weeks... my only regret is that I didn't replace my old printer a year ago.❞
L.A.R., 29 Palms, CA

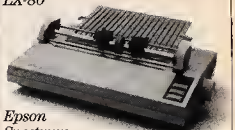
Epson dot matrix printers have very graphic advantages. Besides the versatility of draft and Near Letter Quality and the corresponding speed options, they can draw some very convincing pictures, including true circles.

Recently in Martinez, California, two cats knocked an Epson printer 4 feet onto a tile/concrete floor. Although it didn't land on its feet, the printer is still running.

NOW THAT YOU CAN AFFORD AN EPSON, WHAT ARE YOU WAITING FOR?



Epson Spectrum LX-80



Epson Spectrum LX-90

For more details, call (800) 421-5426 or in California (213) 539-9140.

EPSON

Number One.
And built like it.

Epson is a registered trademark of Epson Corporation. Apple, Apple IIc and Apple IIe are registered trademarks of Apple Computer, Inc. IBM, IBM PC and IBM PCjr are registered trademarks of International Business Machines Corporation. PIC and Spectrum are trademarks of Epson America, Inc.

COMPUTE!

SEPTEMBER 1985
VOLUME 7
NUMBER 9
ISSUE 64

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 70 before typing in programs.

AP Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer.
*General Interest.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 10954, Des Moines, IA 50340. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1985 by COMPUTE! Publications, Inc. All rights reserved, ISSN 0194-357X.

Editors Notes

Last month we mentioned some apparent communication problems regarding access to the new Amiga from Commodore. We're happy to report that comments in our editorial became moot before they reached print. Commodore's new senior management team moved quickly and smoothly to see that we, along with other magazines in the industry, received even-handed treatment in access to information.

The Amiga is an important product. We see a significant, lasting change in the way personal computers will be used and programmed and, thus, in the ways we cover computers. With the introduction of the Amiga (see the story on page 16) and the ST from Atari, consumer computing will never be the same again.

Among other things, BASIC now faces its first serious challenge as the language of popular computing. When you turn on these new computers, you don't see the familiar BASIC greeting "READY." Instead, you see a Macintosh-like "desktop" screen with icons, etc. This manager is called Intuition on the Amiga, GEM on the ST. BASIC is only one of several options, several languages you could load into the computer from disk. A simple command, however, exits this environment and lands you in an IBM PC-like AmigaDOS, said to be quite like Unix, an operating system first developed for large minicomputers. The Atari ST's TOS will be similar. Both are command-rich systems, nearly languages in themselves.

COMPUTE! expects to continue to publish the majority of its programs in BASIC. The new machines' BASICs are large and fast. They include a generous set of graphics and sound instructions. Above all, everyone who buys an ST or an Amiga will have BASIC. That language is being shipped with, though not built into, these computers.

Interestingly, most commercial software announced so far for the ST and Amiga is not being written in machine language. Instead, it is being written in C, a language popular among professional programmers which has a reputation for portability between computers. Some have argued that this spells the end of assemblers, the end of writing machine language programs. We do not find that argument compelling.

The argument goes like this: The new machines are faster (because the microprocessor, the 68000, is more efficient) and thus maximizing speed of execution by using machine language is no longer necessary. Compiled languages like C run sufficiently quickly. *Lotus 1-2-3* is written in C. Also, some new BASICs and operating systems are largely C.

The other factor in favor of machine language, its conservation of memory, is now less critical, too. Compilers can use up computer memory rapidly. Amiga BASIC, written mostly in C, is about 96K large; Commodore 64 BASIC, written entirely in machine language, uses up only 8K. Instead of having to fit everything into 64K, the maximum memory which can be easily accessed by the older 8-bit chips, the new computers can access megabytes of memory. Tecmar, an Ohio company, is developing an expansion board for the Amiga which adds up to two megabytes of memory. Hence, bulky, compiled programs don't cause much of a problem. There's memory to spare. However, even though the Amiga and ST each have 192K of ROM space, both machines' operating systems—written largely in C—have to be supplied on disk with early models. The compiled C is too big to be built into ROM until programmers can optimize and condense the code.

C has its advantages, but one fact is overlooked: Machine language is the computer's language. All other languages are compromises, less direct ways of telling the computer what you want it to do. This indirection slows the computer down for many of the same reasons that you would be slowed down in a foreign country. No matter how similar the two languages, from time to time you would be forced to resort to hand signals, symbols, even to looking things up in a dictionary. Likewise, a compiled programming language results in a more or less indirect communication with the computer. Even the best compilers produce bulkier and less efficient programs than does pure machine language.

Something similar to the current popularity of C happened when home computers were first introduced. BASIC was then the most common language for commercial programs. Spreadsheets, word processors, and games were sold which were entirely BASIC. They were slow, had few features, and used up much of the available memory space.

Now that there is a transition from 64K to 512K, quadruple the processing speed, and far better graphics and sound—most any good program is going to be impressive. The new machines make their software look good in the same way that calculators made the early 8K Commodore PET look good. It's a whole new level of power and control. But the shock of the new doesn't last. Software companies will compete along the classic lines: They will all try to offer the fastest product with the most features. Once again we are likely to see a migration to machine language as programmers vie with each other to take their machines to the limit.

The 68000 is not a new chip, but it is new to home computers. Introduced by Motorola in 1981, it cost over $200 until recently. It is the chip in the Apple Macintosh, and sales of that computer have helped drive down the price to its current $20, making it affordable as the new consumer CPU. How does the 68000 differ from the 6502, the chip in most current popular computers (Apple, Atari, Commodore, etc.)? Essentially, things like multiplying large numbers are easier to do, fetching and storing is faster and more efficient, what took several steps to accomplish in the 6502 can now be done in a single operation.

Of course, we won't see the ultimate software the minute the new hardware is introduced. It will take time for programmers to investigate the new territory. But judging from the preliminary software we've seen, the new computers offer stunning opportunities for creative programming and—whatever languages are used—the resulting software will take us far beyond what we've experienced on today's home computers. We plan to bring you some of that stunning programming in the pages of COMPUTE! in the coming years.

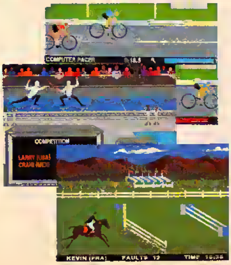
Robert Lock

Editor in Chief

Richard Mansfield

Senior Editor

SUMMER GAMES II.™ EIGHT NEW WAYS TO GO FOR THE GOLD.



Sure Summer Games was great, but why stop there? Let Summer Games II take you even farther with eight new events including cycling, fencing, kayaking, triple jump, rowing, high jump, javelin and even equestrian. They can all be played by up to eight players and some, like cycling, rowing and fencing challenge you with realistic head-to-head competition.

First, you decide which of the 18 different countries you're going to represent. Then, in true Olympic fashion, you will need the proper strategy and mental toughness, not just speed and agility to excel in each event. It's so realistic, there's even an opening and closing ceremony along with medal presentations after each event.

It's not too early to get ready for 1988. With the right diet, proper training and hours of practice you just might make it. In the meantime, put on your sweatsuit, grab that joystick and let Summer Games II give you eight new ways to **Go For The Gold!**

EPYX™
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

Publisher	Gary R. Ingersoll
Editor In Chief	Robert C. Lock
Director of Administration	Alice S. Wolfe

Senior Editor	Richard Mansfield
Managing Editor	Kathleen Martinek
Editor	Tom R. Halfhill
Assistant Editor	Philip Nelson
Production Director	Tony Roberts
Production Editor	Gail Cowper
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Ottis R. Cowper
Assistant Technical Editors	John Krause, George Miller
Program Editor	Charles Brannon
Features Editor	Selby Bateman
Assistant Editor, COMPUTE!'s GAZETTE	Todd Heimarck
Feature Writer	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Kevin Mykytyn, Kevin Martin
Submissions Reviewer	Mark Tuttle
Programming Assistant	David Florance
Copy Editors	Joan Rouleau, Ann Davies
Executive Assistant	Susan Young
Administrative Assistants	Julia Fleming, Iris Brooks, Jan Kretlow
Associate Editors	Jim Butterfield Toronto, Canada Harvey Herman Greensboro, NC Fred D'Ignazio Roanoke, VA David Thornburg Los Altos, CA
Contributing Editor	Bill Wilkinson

COMPUTE!'s Book Division	
Editor	Stephen Levy
Assistant Editors	Gregg Keizer, J. Blake Lambert
Administrative Assistant	Laura MacFadden
Director, Book Sales & Marketing	Steve Voyatzis
Assistant	Carol Dickerson

Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Debbie Bray, Dabney Ketrow
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair

Director of Advertising Sales	Ken Woodard
Production Coordinator	Patti Stokes
Administrative Assistant	Kathleen Hanlon

Promotion Assistant	Caroline Dark

Customer Service Manager	Philippa King
Dealer Sales Supervisor	Gail Jones
Assistants	Debi Goforth, Liz Krusenstjerna, Rhonda Savage
Individual Order Supervisor	Judy Taylor
Assistants	Betty Atkins, Gayle Benbow, Mary Hunt, Jenna Nash, Chris Patty
Warehouse Manager	Lonnie Arden
Staff	Harold Ayers, Steve Bowman, Larry O'Connor, David Hensley

Data Processing Manager	Leon Stokes
Assistant	Chris Cain

Vice President, Finance & Planning	Paul J. Megliola
Director, Finance & Planning	R. Steven Vetter
Accountant	Robert L. Bean
Financial Analyst	Karen K. Rogalski
Staff	Dale Branch, Jill Pope

Credit Manager	Barry L. Beck
Staff	Sybil Agee, Anne Ferguson, Pat Fuller, Doris Hall, Linda Miller, Mary Waddell, Jane Wiggs

Purchasing Manager	Greg L. Smith

Gary R. Ingersoll, President
Paul J. Megliola, Vice President, Finance and Planning
Debi Nash, Executive Assistant
Anita Armfield, Assistant

MPA
Magazine Publishers Association

Member
ABC
Audit Bureau of Circulations

Coming In Future Issues

Save With Replace: Debugged At Last

In-Depth Review: Atari 520ST

Amiga's Amazing Graphics

The Witching Hour: Strategy Game For Commodore, Atari, Apple, IBM, and TI

Apple II Pull-Down Menus

Lightning Renumber For Atari

64 Multicolor Graphics Made Easy

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA
Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA
Telephone: 919-275-9809

Subscription Orders

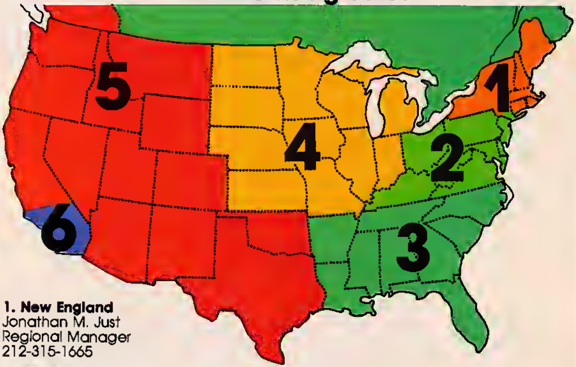
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign 'Surface Mail		$30
Foreign Air Delivery		$65

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Saval
Eastern Advertising Manager
212-315-1665
Kathy Hicks
Marsha A. Gitteiman
215-646-5700
Brian S. Rogers
212-674-0238

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/Mountain/Texas
Phoebe Thompson
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue,
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 914, Farmingdale, NY 11737. Include a note indicating your preference to receive only your subscription.

Authors of manuscripts warrant that all materials submitted to COMPUTE! are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE!, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of COMPUTE! Publications, Inc. No portion of this magazine may be reproduced in any form without written permission from the publisher. Rights to programs developed and submitted by authors are explained in our author contract. Unsolicited materials not accepted for publication in COMPUTE! will be returned if author provides a self-addressed, stamped envelope. Programs (on tape or disk) must accompany each submission. Printed listings are optional, but helpful. Articles should be furnished as typed copy (upper- and lowercase, please) with double spacing. Each page of your article should bear the title of the article, date and name of the author. COMPUTE! assumes no liability for errors in articles or advertisements. Opinions expressed by authors are not necessarily those of COMPUTE!.

TIRED OF WAITING FOREVER FOR YOUR PROGRAMS TO LOAD?

INTRODUCING THE FAST LOAD CARTRIDGE FROM EPYX.

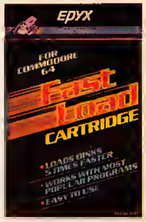
You're tired of waiting forever for your Commodore 64 programs to load. But it's no use glaring at your disk drive. Calling it names won't help, either. It was born slow — a lumbering hippo. You need the FAST LOAD CARTRIDGE from EPYX. FAST LOAD transforms your Commodore 64 disk drive from a lumbering hippo into a leaping gazelle. With FAST LOAD, programs that once took minutes to load are booted up in a matter of seconds.

FAST LOAD can load, save and copy your disks five times faster than normal. It plugs into the cartridge port of your Commodore 64 and goes to work automatically, loading your disks with ease. And that's only the beginning. You can copy a single file, copy the whole disk, send disk commands, and even list directories without erasing programs stored in memory.

And unlike other products, the FAST LOAD CARTRIDGE works with most programs, even copy protected ones, including the most popular computer games.

The FAST LOAD CARTRIDGE from Epyx. Easy to insert, easy to use and five times faster. So why waste time waiting for your disks to load?

Speed them up with FAST LOAD!


EPYX
FOR COMMODORE 64
Fast Load
CARTRIDGE
• LOADS DISKS 5 TIMES FASTER
• WORKS WITH MOST POPULAR PROGRAMS
• EASY TO USE

EPYX™

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, *P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.*

Relational Operators

I recently typed in the TI-99/4A game "Circus" (COMPUTE!, February 1984) and noticed the following statement in line 50:

```
SC=SC+(H=120)*-50+(H=112)*-7
5+(H=104)*-100+((H=128)*(M1=
1)*250)
```

How does this statement work?

Dan Schwarz

Although your question concerns a TI program, the answer applies to BASIC programming on a wide variety of computers. The complex statement that has you puzzled calculates the game score (variable SC) by using the equal sign (=) as a relational operator. *Though its syntax looks odd, it efficiently takes the place of several IF-THEN statements.*

In "Circus" the balloon (variable H) popped by the clown can be in the bottom row (character number 120), in the middle row (character 112), or the top row (104). Character 128 signifies the bonus balloon. A bottom row balloon scores 50 points, the middle row scores 75, the top row is worth 100, and a bonus balloon scores 250 points provided its color is yellow (M1=1; see line 80 of the program).

The expression (H=120) doesn't change the value of H. Instead, it performs a logical test similar to IF. When H equals 120—when you pop a bottom-row balloon—this expression returns a value of −1. Any expression that evaluates to −1 is considered to be true. When H equals any other number, the computer returns 0 to show the expression is false. (TI, Commodore, and IBM PC/PCjr computers evaluate true expressions to −1; Apple, Atari, and Timex/Sinclair computers use 1 rather than −1.)

Say that the clown pops a balloon in the bottom row. Since H equals 120, the expression (H=120) is true and evaluates to −1. This value is multiplied by −50 to add 50 to the score (multiplying two negative numbers produces a positive number). Since H=120 is true, the other expressions (H=112, H=104, and H=128) are false, so the multiplications yield 0 and the score doesn't change. The remaining expressions in the example increment the score when you pop balloons in the middle and upper rows or pop the bonus balloon (character 128) when it's yellow. Other relational operators include <, >, AND, OR, and NOT (if available in your dialect of BASIC). String expressions work as well as numeric expressions, and relational operations are particularly efficient when combined with ON-GOTO or ON-GOSUB statements.

Atari Tape-To-Disk Transfer

When I bought a disk drive for my Atari system, I was faced with retyping all the machine language programs (like *SpeedScript*, COMPUTE!, May 1985) I had previously saved on tape. Instead, I found a way to use "Atari MLX" to load a machine language program from tape, and then either save it as a binary disk file or make a boot disk. To make a binary file, change line 390 of MLX as follows:

```
390 IF N=-19 THEN MEDIA= ASC("
   D"):DTYPE=70:GOTO 720
```

Change line 390 as follows to make a boot disk:

```
390 IF N=-19 THEN MEDIA= ASC("
   D"):GOTO 720
```

After that's done, run MLX and follow the instructions, loading from tape and saving to disk when appropriate.

David L. Pettite

Thank you for the information. Readers should note that this temporary change to line 390 is only for converting tape files to disk files. It is not a correction to MLX, and should not be permanently incorporated into your copy of Atari MLX.

64 Key Beeper

Is there a program for the Commodore 64 that will cause a beep when a key is pressed?

Jeffrey Gurr

The following program adds audible feedback to the keyboard of your 64, as found on Atari computers. (Ironically, owners of Atari 400s and 800s frequently write us for a way to turn off *the built-in keyboard beep.) The program puts a short, interrupt-driven machine language routine in an unused memory area (679–760), activates the beep routine, then erases itself. Be sure to save a copy of the program* before *running it, and turn up the volume on your TV or monitor. This routine is designed to be used in direct mode (while you're typing a program, etc.) rather than in program mode (while a program is running). It doesn't interfere with most BASIC operations, but any program that creates other sounds, changes the hardware interrupt vector, or alters locations 3–4 and 679–760 may disrupt the beep or cause other problems. You should always disable the beep (press RUN/STOP–RESTORE) before running other programs. Enter SYS 679 to turn it back on.*

```
1 S=679:N=S
2 READQ:IFQ=256THEN4
3 POKEN,Q:N=N+1:CK=CK+Q:GOTO2
4 IFCK<>9233THENPRINT"ERROR IN
   DATA":END
5 SYS(S):NEW
6 DATA 120,169,206,141,20,3,16
  9,2,141,21,3
7 DATA 162,0,138,157,0,212,232
  ,224,25,208,248
8 DATA 169,15,141,24,212,169,6
  7,141,5,212,169
9 DATA 17,141,1,212,88,96,165,
  197,201,64,240
10 DATA 30,197,3,208,6,165,4,2
   40,2,208,24
11 DATA 169,32,141,4,212,169,3
   3,141,4,212,165
12 DATA 197,133,3,169,1,133,4,
   208,4,169,0
13 DATA 133,4,76,49,234,256
```

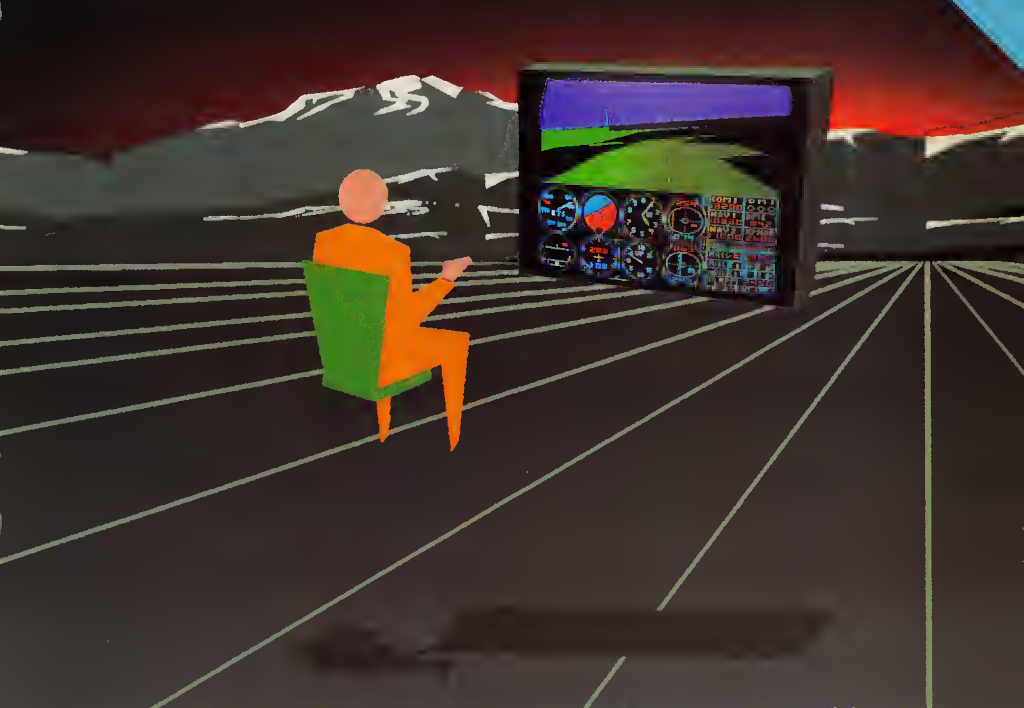
Simpler IBM Unprotection

On CompuServe's PC-SIG disk #184 you can find a simpler procedure for unlocking protected IBM BASIC programs (see "Unlocking IBM BASIC Programs" by Peter Nicholson, COMPUTE!, June 1985). Written by Todd Pollock, this method uses BSAVE and BLOAD commands to restore the portion of RAM that is disabled by a protected program. First, type in any two- or three-line BASIC program such as this:

```
10 PRINT "HELLO"
```

Flight Simulator II

For Apple, Atari®, & Commodore 64™



Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders enclose $49.95 plus $2.00 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800 / 637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

```
20 GOTO 10
30 END
```

Save the program by entering this line: BSAVE "UNPRO.CIM",&H400 ,&H7F. To unprotect a protected program, load the protected program into memory, then enter this line: BLOAD "UNPRO.CIM". I suspect that Nicholson's procedure may be required on some compatibles, since Pollock's does not simply query a standard location for standard information. A quick test on my friend's Sperry PC-compatible showed that it disables the BLOAD command while a protected program is in memory. However, Pollock's procedure does have the advantage of requiring much less typing.

Guy R. Winters

We tested this method on the PC and PCjr and found that you need to BSAVE only one byte of memory. Type in any one-line program such as 10 END. Then enter this command: BSAVE"UN.PRO",1124,1. The BSAVE command saves one byte of memory at location 1124 (&H464 hexadecimal). Now load a protected program (one that was saved with SAVE"filename",P), and load the one-byte file with BLOAD"UN-.PRO". On the PC/PCjr, the protection evaporates and you can list, edit, or save the program as usual. Also, PEEK and POKE are reenabled in direct mode.

The PC and PCjr use location 1124 as a flag: It contains 0 when an unprotectea program is in memory and 254 after you load a protected program. The BSAVE shown above saves location 1124 at a time when we know the flag is set to 0. The BLOAD simply loads the 0 back into location 1124, resetting the flag to signify no protection. As you found by testing your friend's Sperry, "compatibility" is a relative concept. Evidently one of the Sperry designers knew or anticipated this trick, and prevented it by disabling BLOAD.

Although program protection disables POKE and PEEK in immediate mode, both commands are still legal in program *mode (at least on the PC/PCjr). Thus, a protected program can unprotect itself while running (for instance, if you enter a password) and an unprotected program can protect itself as well. The PCs we tested put a 254 in location 1124 to indicate protection, but in fact any nonzero value seems to set the protection flag: Editing, listing, PEEKing, and POKEing are ruled out, and you can resave the program only in protected format.*

Disabling Apple's Break Key

According to your answer to Alex Tarlecky's letter in December 1984, the RESET key can be disabled on the Apple IIc with the command POKE 1012,PEEK(1012) AND 10. But is there a way to also disable the CONTROL-C function to keep people from breaking out of my programs?

Mike Sanders

Yes, there is. After Applesoft BASIC executes a program statement, it checks for any errors that might have occurred. At the same time, it checks to see if CTRL-C was pressed. If so, Applesoft responds as it does when it encounters a syntax error or illegal quantity error. Normally, it stops the program and displays an appropriate error message (BREAK IN line#).

The secret to trapping CTRL-C is an instruction that changes the way Applesoft handles such errors—the ONERR statement. For instance, once the computer executes a statement such as ONERR GOTO 1000, it responds to any error—including the CTRL-C function—by transferring control to line 1000 (or any other line you specify with ONERR). Make sure, however, that the line specified in the ONERR statement actually exists in your program. Otherwise, Applesoft searches for an undefined line when an error happens, causing another error. The result is an endless loop and a locked-up computer.

You should put an error-handling routine starting at the line number referred to by ONERR. This routine should PEEK location 222, which contains an error code. If this location contains 255, then CTRL-C was pressed. The best way to deal with CTRL-C is to have your error routine GOTO the program's main menu or some other predictable location, so that CTRL-C still causes a break but doesn't stop the program.

If PEEK(222) isn't 255, then CTRL-C wasn't *pressed—an actual error occurred. This could be a disk error (wrong disk in the drive, no disk, disk full, etc.) or an error in your program. It is usually easier to let Applesoft handle the errors that you aren't expecting. You can do this by POKEing memory location 216 with 0 to cancel the ONERR trap. Then use the Applesoft RESUME instruction, which reexecutes the statement that caused the error in the first place. Since the instruction didn't finish the first time, you should get the same error, but this time the program halts with an appropriate error message.*

TI Supplies

Just after I purchased a TI-99/4A computer, the company went out of business. Does this mean I won't be able to purchase anything for my computer? I would like to purchase Extended BASIC, a printer, and other peripherals.

Kathy Armstrong

Texas Instruments is still very much in business; it has simply stopped manufacturing home computers such as the TI-99/4A. Fortunately, TI-99/4A products are still available. The following firms carry software, hardware, and peripherals (this is the most complete and accurate list we were able to compile at time of publication):

Triton Products
P.O. Box 8123
San Francisco, CA 94128
1-800-227-6900

Unisource Electronics, Inc.
P.O. Box 64240
Lubbock, TX 79464
1-800-858-4580

MSW Computers & Electronics
22 East Tioga Street
Tunkahannock, PA 18657
1-800-233-3266

Tenex Computer Express
P.O. Box 6578
South Bend, IN 46660
219-259-7051

Reader Cynthia Becker informs us that hardware and software are also available through the TI-99/4A National Assistance Group. After paying a $10 membership fee, you are entitled to purchase TI products from this organization and receive its newsletter as well:

TI-99/4A National Assistance Group
P.O. Box 290812
Ft. Lauderdale, Florida 33329
(305) 583-0467

Commodore 16 Conversions

I have found that programs written for the VIC-20 Super Expander will run on the Commodore 16 as well if you add the BASIC 3.5 statement SCALE 1=1023*1023 to the beginning of the program. The 16 uses different tokens for graphics keywords like DRAW, POINT, and so on. But the programs will load without any problem from disk or tape. After you load the program, edit the lines that contain those keywords and save it again. It should run just fine.

John Elliot

Thanks for the information.

Trapping IBM's Break Key

I own an IBM PC and have been trying to trap the Ctrl-Brk keys. I have looked in a tremendous number of books, but still couldn't find anything about it. I haven't been able to scan the keyboard for the information I need. How can I trap those keys?

Patrick McGarry

Since many readers have asked this question, we'll show you two techniques that work with BASICA or Cartridge BASIC on either the PC or PCjr. The following program traps both Ctrl-Break (break) and Ctrl-Alt-Del (reboot).

CLOSE ENCOUNTERS OF THE FANTASY KIND

QUESTRON
A Fantasy Adventure Game
SSI
STRATEGIC SIMULATIONS INC
Action, strategy, & adventure in the netherworld
GEMSTONE WARRIOR
SSI
STRATEGIC SIMULATIONS, INC

QUESTRON™ *A role-playing game so spell-binding, it suspends reality and conjures up an enchanted world whose only hope against the forces of evil is you.*

GEMSTONE WARRIOR™ *A lightning-fast action strategy game that will leave you breathless with excitement as it takes you through the labyrinths of the Netherworld on a noble quest.*

Open them up and step through the gateway to your fantasies come true. At your local computer/software or game store today.

QUESTRON™ is available on 48K diskette for APPLE® II with Applesoft ROM, Apple II+, IIe, and IIc ($49.95). On 40K diskette for ATARI® home computers ($49.95). On 64K diskette for COMMODORE 64™ ($39.95).

GEMSTONE WARRIOR™ is available on 48K diskette for APPLE® II with Applesoft ROM, Apple II+, IIe, and IIc ($34.95). Also on 64K diskette for COMMODORE 64™ ($34.95).

SSI™

© 1985 by **STRATEGIC SIMULATIONS, INC.** All rights reserved.

If there are no convenient stores near you, VISA & M/C holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, 800-772-3545, ext. 335. Please specify computer format and add $2.00 for shipping and handling.

To order by mail, send your check to: STRATEGIC SIMULATIONS INC, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 7% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

APPLE, ATARI and COMMODORE 64 are trademarks of Apple Computer, Inc., Atari, Inc., and Commodore Electronics, Ltd., respectively.

```
10 CLS:PRINT "Try to use Brea
   k or Ctrl-Alt-Del"
20 B$=CHR$(4)+CHR$(70):C$=CHR
   $(12)+CHR$(83)
30 KEY 15,B$:KEY (15) ON:ON K
   EY (15) GOSUB 80
40 KEY 16,C$:KEY (16) ON:ON K
   EY (16) GOSUB 90
50 FOR J=1 TO 9999:NEXT:PRINT
    "Break & Ctrl-Alt-Del wor
   k now"
60 KEY (15) OFF:KEY (16) OFF
70 GOTO 70
80 PRINT "Break has no effect
    right now.":RETURN
90 PRINT "Rebooting is a very
    bad idea.":RETURN
```

Once the key trap is set (lines 20–40 above), the system checks for a trap between every statement of the main program. When the right keys are pressed, execution diverts immediately to the trapping subroutine, no matter what the main program is doing at the time. Since the trap can be sprung between any two statements in the program, strange results may occur if you don't anticipate the possible diversion. Of course, the trapping subroutine doesn't have to print a message (or do anything else except end with RETURN). You can also disable Break by changing the computer's break interrupt vector at locations 108–112 (&H6C–&H6F), as shown here:

```
10 DEF SEG=0:FOR J=0 TO 3:A(J
   )=PEEK(108+J):NEXT
20 POKE 108,64:POKE 109,1:POK
   E 110,112:POKE 111,0
30 PRINT "Try to use Ctrl-Brk
    (PC) or Fn-Brk (PCjr)
40 FOR J=1 TO 9999:NEXT:PRINT
    "Brk key works again"
50 FOR J=0 TO 3:POKE 108+J,A(
   J):NEXT
60 GOTO 60
```

This program diverts the system's normal break routine to a do-nothing IRET (return) instruction in ROM (Read Only Memory). Don't forget to restore the normal vector when the program ends (line 50). These examples are drawn from Russ Davies' Mapping the IBM PC and PCjr *(published by COMPUTE! Books), which contains additional information on keyboard programming from DOS and machine language.*

Commodore ML Addresses

I own a Commodore 64. How can I find the beginning and ending addresses of a machine language program stored on disk?

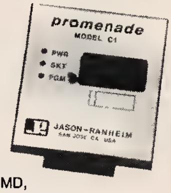
Eric Adams

The following program does the job on any Commodore computer with a disk drive (except the 128 in CP/M mode). The first two bytes of a disk program file contain the load address in low byte/high byte format. This program finds the beginning, then reads to the end of the file. The end address equals the start address plus the number of bytes read. (Of course, a disk data file—which holds data rather than a program—has no load address.)

```
1 INPUT"FILENAME";F$:A$="0:"+F
  $+",P,R":OPEN 2,8,2,A$
2 GET#2,A$:GOSUB 5:L=A:GET#2,A
  $:GOSUB 5:SA=L+256*A:PRINT"S
  TART";SA
3 GET#2,A$:IF ST=0 THEN SA=SA+
  1:GOTO 3
4 PRINT"END";SA:CLOSE 2:END
5 IF A$="" THEN A$=CHR$(0)
6 A=ASC(A$):RETURN
```

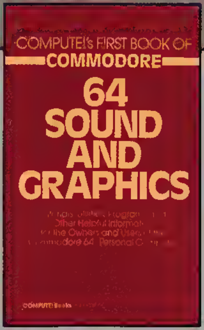
Tape users can find beginning and ending addresses with only two program lines. The following routine runs as listed on the Commodore 64, VIC-20, and PET. Plus/4 and 16 users should subtract 10 from the four addresses in line 2 (replace 829 with 819, 830 with 820, and so on). Commodore 128 users (in 128 mode) should replace the same four addresses with 2817, 2818, 2819, and 2820. The header data stored at the beginning of a tape file contains the program's starting and ending addresses. The method shown here simply OPENs the file to read the header into the tape buffer, then PEEKs the addresses from the buffer.

```
1 INPUT"FILENAME";F$:OPEN 2,1,
  0,A$:CLOSE 2
2 PRINT"START";PEEK(829)+256*P
  EEK(830);CHR$(13);"END";PEEK
  (831)+256*PEEK(832)
```
©

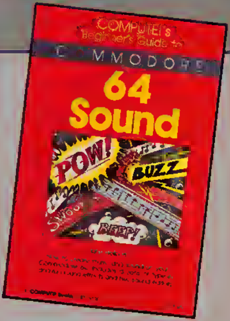


GREAT Sound and Graphics

COMPUTE!'s Reference Guide to Commodore 64 Graphics, John Heilborn, $12.95
A complete tutoriol on creating graphics displays, pictures, ond animation on the Commodore 64.



COMPUTE!'s First Book of Commodore 64 Sound and Graphics, Edited, $12.95
Clear, useful explanations of the 64's sound and graphics capabilities including tutorials and exomple programs.



COMPUTE!'s Beginner's Guide to Commodore 64 Sound, John Heilborn, $12.95
Complete descriptions and numerous examples show you how to program sound ond music on the 64.

COMPUTE! Books offers you three fascinating, informative books on sound and graphics for your COMMODORE 64 at a special savings.

Buy all three books for only $29.00.
That's 25 percent off the retail price!

Limited Time Only

These books give you up-to-date, easy-to-understand tutorials, programs, and examples to help you maximize the sound and graphics features on your Commodore 64. Order today to receive all three books for only $29.00. This offer expires October 10, 1985.

Order directly from COMPUTE! Books by moiling the ottoched form with your poyment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403. For the fastest service, coll toll-free 800-334-0868 (in NC coll 919-275-9809).

Order Form

Pleose send me:

_____ *COMPUTE!'s Beginner's Guide to Commodore 64 Sound* (54-X), $12.95 _____

_____ *COMPUTE!'s First Book of Commodore 64 Sound ond Graphics* (21-3), $12.95 _____

_____ *COMPUTE!'s Reference Guide to Commodore 64 Graphics* (29-9), $12.95 _____

_____ SPECIAL SAVINGS—ALL THREE BOOKS FOR $29.00 _____

Subtotol _____

NC residents add 4.5% tox _____

Shipping charges* _____

*($2.00 per book in U.S. and surface mail; outside U.S. $5.00 airmail per book)

Totol paid _____

ALL ORDERS MUST BE PREPAID IN U.S. FUNDS

□ Poyment enclosed (check or money order)

□ Charge □ MosterCord □ Viso □American Express

Account No. _____ Exp. Date _____
(Required)

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4-6 weeks for delivery 7591171

COMPUTE! Books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

AMIGA HR
AMIGA

The AMIGA: An In-Depth Review

Tom R. Halfhill, Editor

Three years in the making, Commodore's new Amiga personal computer was finally introduced at a lavish media event in New York this summer. Commodore says the new machine should be available by the end of August. This report was compiled from sessions with the Amiga prior to its release.

Commodore's Amiga is much more than just another new computer. It's a pivotal machine that may well shatter the traditional boundaries and prejudices which for years have divided the microcomputer marketplace. It defies classification as simply a home computer, game computer, business computer, or hacker's computer. In fact, the Amiga's power, versatility, and ease of use may qualify it as the first true personal computer.

The Amiga is not a me-too clone, or a cautious step sideways, or an incremental step forward. It's a genuine leap to a new generation of advanced personal computers. The Amiga will be the yardstick by which all other new computers over the next few years will be measured.

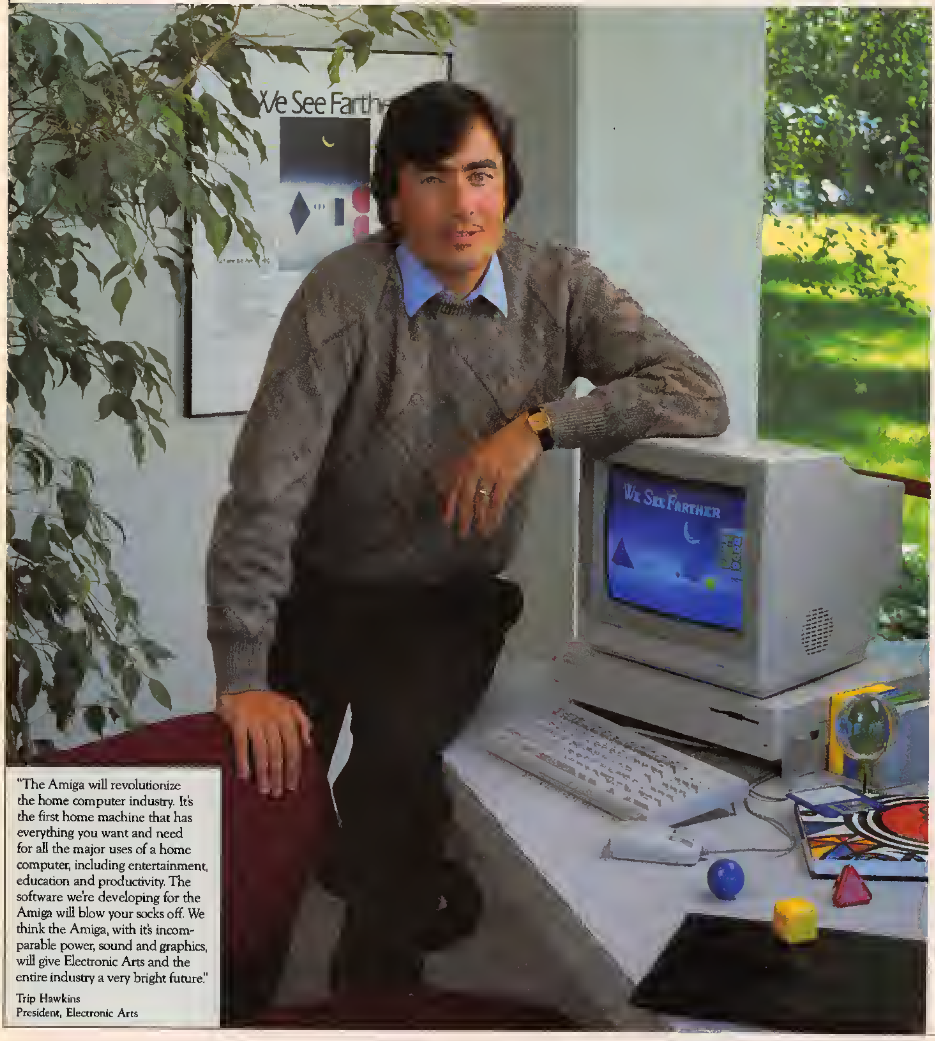
What sets the Amiga apart is that no other computer on the market can do so many things so well. To match its power as a business computer, you'd have to go all the way to a $4,000 IBM AT or even a minicomputer; to surpass its graphics and animation capabilities, you'd have to invest in a $10,000 dedicated graphics terminal; to surpass its sound and music features, you'd have to buy a music synthesizer. The Amiga is that rare example of a general-purpose machine that excels at specialized applications.

This versatility transcends the traditional computer categories taken for granted over the years. For example, although it's certainly possible to use a machine such as a Commodore 64 as a business computer, or a machine such as an IBM PC as a home computer, some compromises are usually inevitable. But the Amiga should prove to be equally suitable for the most demanding business people, home users, programmers, educators, children, video artists, and electronic musicians. In addition, it's easy enough for a beginner to learn quickly, yet deep enough to fascinate the most impassioned latenight hacker.

Commodore, too, senses that it has a new kind of computer on its hands. The company is going out of its way to avoid calling the Amiga a business computer or a home computer. Furthermore, Commodore is

A message from a leading software publisher.

WHY ELECTRONIC ARTS



"The Amiga will revolutionize the home computer industry. It's the first home machine that has everything you want and need for all the major uses of a home computer, including entertainment, education and productivity. The software we're developing for the Amiga will blow your socks off. We think the Amiga, with it's incomparable power, sound and graphics, will give Electronic Arts and the entire industry a very bright future."

Trip Hawkins
President, Electronic Arts

IS COMMITTED TO THE AMIGA.

In our first two years, Electronic Arts has emerged as a leader of the home software business. We have won the most product quality awards—over 60. We have placed the most *Billboard* Top 20 titles—12. We have also been consistently profitable in an industry beset by losses and disappointments.

Why, then, is Electronic Arts banking its hard won gains on an unproven new computer like the Amiga?

The Vision of Electronic Arts.

We believe that one day soon the home computer will be as important as radio, stereo and television are today.

These electronic marvels are significant because they bring faraway places and experiences right into your home. Today, from your living room you can watch a championship basketball game, see Christopher Columbus sail to the New World, or watch a futuristic spaceship battle.

The computer promises to let you do much more. Because it is interactive you get to participate. For example, you can play in that basketball game instead of just watching. You can actually be Christopher Columbus and feel firsthand what he felt when he sighted the New World. And you can step inside the cockpit of your own spaceship.

But so far, the computer's promise has been hard to see. Software has been severely limited by the abstract, blocky shapes and rinky-dink sound reproduction of most home computers. Only a handful of pioneers have been able to appreciate the possibilities. But then, popular opinion once held that television was only useful for civil defense communications.

A Promise of Artistry.

The Amiga is advancing our medium on all fronts. For the first time, a personal computer is providing the visual and aural quality our sophisticated eyes and ears demand. Compared to the Amiga, using some other home computers is like watching black and white television with the sound turned off.

The first Amiga software products from Electronic Arts are near completion. We suspect you'll be hearing a lot about them. Some of them are games like you've never seen before, that get more out of a computer than other games ever have. Others are harder to categorize, and we like that.

For the first time, software developers have the tools they need to fulfill the promise of home computing.

Two years ago, we said, "We See Farther." Now Farther is here.

ELECTRONIC ARTS™

Dr J and Larry Bird Go One-On-One

The number one software sports game of all time. Shoor as accurately as Larry Bird, slam dunk like the Doctor, while you're cheered on by the victory chants of the Boston Garden crowd.

Skyfox™

Ger in the spaceship and fly. Out your window or on your radar screen you have but split-seconds to appreciate the fierce beauty of enemy jets and tanks

Seven Cities of Gold™

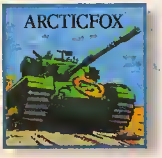
Be Chrisropher Columbus and discover the New World. Learn history and geography, or generate your own random new worlds to explore.

Archon

A new kind of computerized board game, like chess with wizards and dragons for pieces. Bur when one lands on another, they have to fight a white-knuckled action battle.

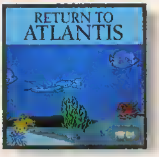
Deluxe Video Construction Set™

Be your own video direcror for business presentations or jusr for fun. Set up special effects, animated computer graphics, sound effects and ritles – even record them to videotape for use with a VCR.



Arcticfox™

You command rhe advanced and deadly tank of the future – the Arcticfox. A first person tank combar game with all the stunning graphics and sound of the best 3-D simulations.



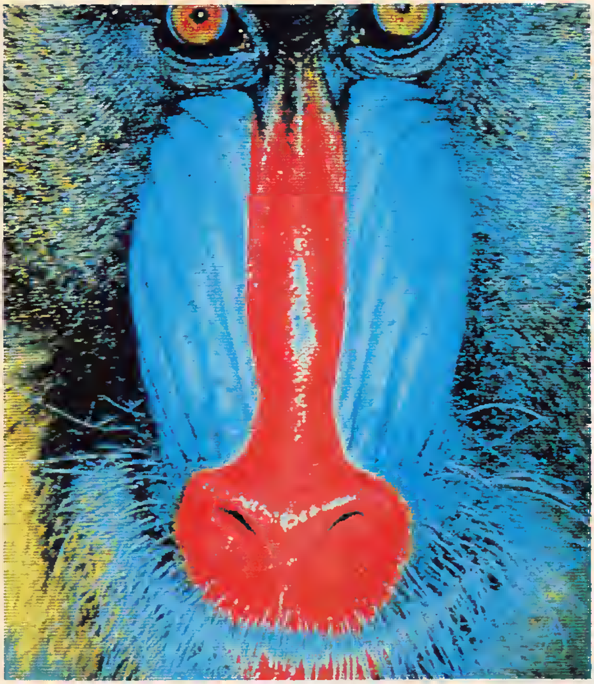
Return to Atlantis™

Play Indiana Cousteau, oceanic hero, in this three dimensional simulation under the seven seas.

Marble Madness™

For the firsr time, the home version of a coin-op arcade game is just as good as the original. Same graphics. Same sound. And you can play it in your bathrobe.

For details about availability, see your Amiga software dealer or call us at (415) 572-ARTS For a product catalog send $.50 and a stamped, self-addressed envelope to: Electronic Arts, Amiga Catalog Offer, 2755 Campus Drive, San Mateo, CA 94403. Amiga is a trademark of Commodore Business Machines Skyfox, Seven Cities of Gold, Deluxe Video Construction Set, Arcticfox, Return to Atlantis and Electronic Arts are trademarks of Electronic Arts Marble Madness is a trademark of Atari Games, Inc



High-resolution graphics on the Amiga are startlingly close to broadcast-quality TV pictures. This image of a mandrill was digitized directly from a photograph and reproduced on the Amiga's 640 × 400-pixel screen.

trying to disassociate the Amiga from its earlier line. The label on the computer, peripherals, and company-branded software says "Amiga," not "Commodore"; and one Commodore executive has asked writers to refer to the computer as the "Amiga from Commodore" rather than the "Commodore Amiga." Apparently, Commodore doesn't want potential buyers to prejudge the Amiga by Commodore's previous products. Although the best-selling VIC-20 and Commodore 64 have earned well-deserved reputations as powerful computers for the price, they are dismissed by some as "game computers" or "toy computers." But now there's an under-$1,500 personal computer which can comfortably outperform much more expensive business computers as well as the best arcade machines.

More than old technology may be rendered obsolete by computers like the Amiga. The new generation may also change a lot of old-fashioned thinking.

Here's a quick review of the Amiga's major features:

• Motorola 68000 chip for the central processing unit. This 16/32-bit microprocessor is also found in the Apple Macintosh and Atari ST series.

• Three special integrated chips nicknamed Portia, Daphne, and Agnes. Portia handles sound and input/output; Daphne handles the video; Agnes controls memory access and also contains two special devices, blitter and copper (short for *coprocessor*), which work together to produce stunning animation and graphics.

• 256K of Random Access Memory (RAM) standard. A clip-on memory board that hides behind a plastic cover on the front of the system unit adds another 256K; further expansion up to six megabytes (6,144K) is possible by adding boards onto the side expansion bus (see below).

• 192K of Read Only Memory (ROM) containing operating system routines. Most of the operating system, however, is loaded from disk into RAM on early model Amigas. This leaves about 130K RAM free on a 256K system. The operating system won't be burned into ROM chips until later. Commodore hasn't decided if upgrade ROMs will be available for early purchasers.

• Built-in microfloppy disk drive. This double-sided drive squeezes 880K of data on a single hardshell 3½-inch disk. Four external drives can be daisy-chained to a port on the back panel.

• Two-button mouse controller. This plugs into one of the two joystick ports on the side of the machine.

• Detached typewriter-style keyboard with separate cursor keys, numeric keypad, and ten special function keys. Interestingly, the keyboard not only returns a value when a key is pressed, but also when the key is released—a highly unusual feature. Also, Commodore says the Amiga can be operated completely from the keyboard, even if you unplug the mouse and hurl it across the room by its wire tail.

• Two-level operating system—AmigaDOS and Intuition, a Macintosh-style user interface that uses a mouse, icons, pull-down menus, screen windows, and multiple screens.

• Multitasking. The Amiga can run several application programs *simultaneously,* and AmigaDOS can even perform several DOS functions at once in different screen windows.

• Four sound channels with stereo output. The sound capabilities are the best of any personal computer available—a wide variety of musical instruments can be simulated with fidelity approaching that of professional-quality synthesizers. A pair of phono jacks on the rear panel sends two sound channels to each auxiliary input jack on your stereo, or they can be plugged into a mono sound system. There are also provisions for digital sound sampling with optional equipment.

This picture was created on the 320 × 200 graphics screen by an artist at Island Graphics, an Amiga software developer.

- Outputs for analog RGB (red-green-blue) monitors, composite color and monochrome monitors, and TV sets. Commodore is selling its own fine-pitch RGB monitor under the Amiga brand name. An RGB monitor is highly recommended for the Amiga, because the higher-resolution graphics modes exceed the capabilities of composite monitors and TVs.
- Centronics-standard parallel port for printers and other peripherals.
- RS-232 serial port for printers, modems, and other peripherals. Tecmar, Inc., of Cleveland, Ohio, is introducing a 2400 bits-per-second modem for this port.
- Expansion port that carries every line on the system bus. This port, on the right side of the system unit, is extremely versatile and will be used for memory expansion beyond 512K RAM, among other things. Tecmar is introducing a 20-megabyte hard disk drive and an expansion board that adds a battery-backed-up clock/calendar, a second RS-232 port, and up to two megabytes of RAM. Coprocessors are another possibility.
- A total of 4,096 colors, far surpassing any other personal computer on the market. Up to 16 or 32 colors can be displayed simultaneously in the standard graphics modes, and all 4,096 can be shown onscreen in a special mode called *hold and modify.*
- Graphics modes of 640×400 with 16 colors; 640×200 with 32 colors; 320×400 with 16 colors; and 320×200 with 32 colors. The screen display system bears a closer resemblance to 8-bit Atari computers than to existing Commodores—not surprising, since some of the Amiga designers were among those who built the original Atari 800 in the late 1970s. For example, a series of memory registers—not color memory—determines which colors will be selected onscreen. Among other things, that means that the 16 or 32 colors displayable in the graphics modes can be any of the 4,096 possible hues, and that changing a color register instantly changes the color of everything previously drawn in that color.
- Eight multicolor sprites. The sprites can be reused on various parts of the screen to create even more moving objects. In some ways, they resemble Atari player/missile graphics instead of Commodore 64-style sprites—they aren't square blocks, but rather tall strips which extend the full height of the screen. Unlike Atari players or Commodore sprites, however, the Amiga's sprites are 16 pixels wide and can display four colors simultaneously with resolution equivalent to the 320 × 200 mode. By overlaying sprites, up to 16 colors can be displayed per object.
- Text modes of 40, 60, or 80 columns. Actually, the Amiga has no true text modes in the conventional sense; all characters are displayed in high-resolution graphics. This makes possible a wide variety of onscreen type styles.
- Speech synthesis as a standard feature. This is simulated in software, not built into the hardware. The male voice seems to have a foreign accent and definitely sounds like a computer, but is more understandable than most speech synthesizers. English text-to-speech conversion is included.
- BASIC on disk. Two BASIC interpreters are in the final stages of development—ABasiC (Amiga BASIC) and a Microsoft BASIC which Commodore says resembles Microsoft BASIC for the Macintosh. According to Commodore, the Amiga will be shipped with the Microsoft BASIC, and ABasiC will be optional. Both are very powerful languages with support for graphics, animation, sound, operating system calls, and the Intuition user interface. Other interpreters, compilers, and assemblers (including Pascal, Forth, and C) will be available soon after the Amiga is introduced.

Although prices still haven't been firmed up at this writing, it appears the basic system unit with 256K RAM, built-in disk drive, detached keyboard, mouse controller, operating system software, and BASIC will cost $1,000 to $1,500. The same system with 512K RAM and a high-resolution RGB color monitor will cost about $2,000.

As personal computers have grown more powerful over the years, designers have wrestled with a dilemma: ease of use versus full flexibility. Beginners and casual users need a computer that's simple to learn and operate, while advanced users don't want to be bogged down with distractions.

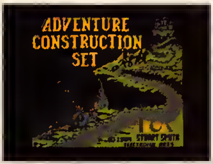
The Amiga designers have worked out a compromise by offering an operating system that can be used both ways. With Intuition, the Macintosh-like interface, you can manipulate the system simply by pointing to menu items or icons representing the functions you want. For example, to call a disk directory on a Commodore 64, you

[This isn't just hype. We really do have the money.]

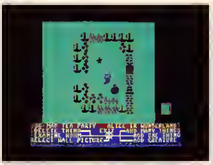
WANTED:

WORLD'S GREATEST ADVENTURES.

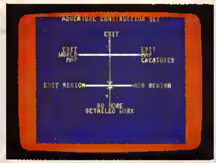
REWARD: $1000 *and* FAME. CALL (800) 227-6703.



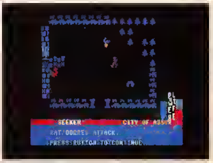
ACS is your tool kit More than 7000 parts to build from. Power to spare!



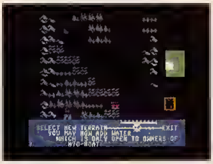
Or make your own parts from scratch. Draw 'em. Give 'em properties. Go for it!



ACS is menu driven. It works with a positive "stick shift" controller. Easy as pie!

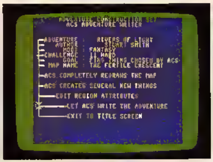


Ready-made segments of sci-fi, mystery & dragon-type adventures included. A real bargain!

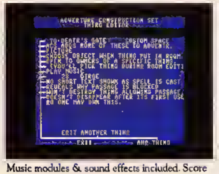


YOU MAY NOW ADD WATER WHICH IS ONLY OPEN TO OWNERS OF H2O-BOAT

Mountains, oceans, forests & etc. are pre-built. Or make your own landscapes. You're the boss!

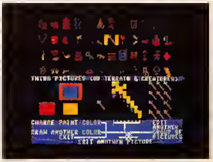

ADVENTURE CONSTRUCTION SET
ACS ADVENTURE WRITER
ADVENTURE : RIVERS OF LIGHT
AUTHOR : STUART SMITH
MOOD : FANTASY
CHALLENGE : 11 HARD
GOAL : THING CHOSEN BY ACS
MAP NAME : THE FERTILE CRESCENT
ACS COMPLETELY REDRAWS THE MAP
ACS CREATES SEVERAL NEW THINGS
EDIT REGION ATTRIBUTES
LET ACS WRITE THE ADVENTURE
EXIT TO TITLE SCREEN

Amazing fact #1: ACS makes adventures (& finishes parts of yours) by itself. This is No Lie!

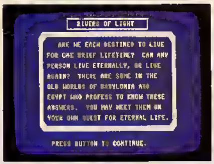


ACS ADDS MORE OF THESE TO ADVENTR.
CHOOSE OBJECT WHEN THING PUT IN ROOM.
REFER TO OWNERS OF A SPECIFIC THING.
(YOU'LL PICK THING DURING ROOM EDIT)
NO SHORT TEXT SHOWN AS SPELL IS CAST
REVEALS WHY PASSAGE IS BLOCKED
WON'T DESTROY THING ALLOWING PASSAGE
DOESN'T DISAPPEAR AFTER ITS FIRST USE
NO ONE MAY OWN THIS.

Music modules & sound effects included. Score your adventures like movies. Be a maestro!

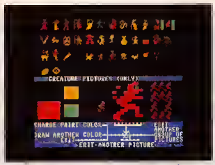

THING PICTURES (OR TERRAIN & CREATURES)
CHANGE PAINT COLOR
DRAW ANOTHER COLOR
EXIT
EDIT ANOTHER PICTURE
EDIT ANOTHER GROUP OF PICTURES

No arms control here! Endless weapons of your design. (Endless armor & cures, fortunately.)

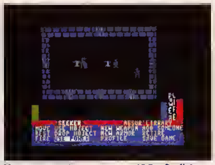


ARE WE EACH DESTINED TO LIVE FOR ONE BRIEF LIFETIME? CAN ANY PERSON LIVE ETERNALLY, OR LIVE AGAIN? THERE ARE SOME IN THE OLD WORLDS OF BABYLONIA AND EGYPT WHO PROFESS TO KNOW THESE ANSWERS. YOU MAY MEET THEM ON YOUR OWN QUEST FOR ETERNAL LIFE.

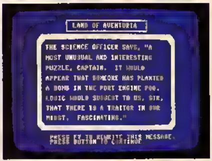
Complete, top-notch epic adventure included: *Rivers of Light* by Stuart Smith, author of *Ali Baba!*



500 obnoxious creatures thrown in! Or make your own. Point 'em at the population! See their priorities!



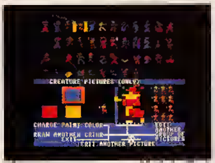
You get magic! 31 variations on 15 Big Spells! You get technology. You get religion. All right!



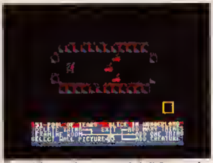
THE SCIENCE OFFICER SAYS, "A MOST UNUSUAL AND INTERESTING PUZZLE, CAPTAIN. IT WOULD APPEAR THAT SOMEONE HAS PLANTED A BOMB IN THE PORT ENGINE POD. LOGIC WOULD SUGGEST TO US, SIR, THAT THERE IS A TRAITOR IN OUR MIDST. FASCINATING."

PRESS F7 TO REWRITE THIS MESSAGE.
PRESS BUTTON TO CONTINUE

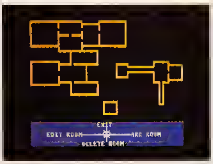
Amazing fact #2: ACS integrates text and graphics. All other products wimp out in this regard.



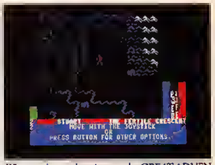
Characters? How many you want? Make 'em. Break 'em. Cast them into the world without a dime!



7 tutorial mini-adventures included! Generously survivable for those tentative first timers!



Design up to 15 different regions with 16 different rooms in each one. Space out!



When you've made a zippo-zocko GREAT ADVENTURE, send it here. You could win 1,000 smackers!

ADVENTURE CONSTRUCTION SET™

from ELECTRONIC ARTS™

HOW TO WIN: Send your adventure entries to Electronic Arts by 12/30/85. 3 winners will be selected by our judges. Their decision: final. Your adventure becomes "public domain" which means you can still get famous even if you don't get rich. **HOW TO ORDER:** Visit your retailer or call 800-227-6703 for direct VISA or MasterCard orders (in CA call 800-632-7979). To buy by mail, send check or money order to Electronic Arts, P.O. Box 306, Half Moon Bay, CA 94019. Add $3 to cover insured shipping and handling. Indicate desired machine version. Allow 1-4 weeks for delivery. *All EA products purchased direct have a 14 day "satisfaction or your money back" guarantee.* **PRICE & MACHINE AVAILABILITY:** C-64 $39.95. Apple IIe/c $49.95. Coming soon to IBM PC/PCjr $49.95. To receive our complete product catalog and direct order form, send 50¢ and a self-addressed, stamped envelope to Electronic Arts, Catalogue Offer, 2755 Campus Drive, San Mateo, CA 94403.

ANNOUNCING THE FAMILY SOFTWARE SPECTACULAR.™

BUY 3, GET 1 FREE.

CHOOSE FROM OVER 50 LEADING TITLES.

What do you call an offer that brings together, for the first time ever, five leading brands of home software?

The Family Software Spectacular!

And it's your opportunity to choose a *free* software program from the best titles on the market today. All you have to do is buy any combination of three titles from any participating brands during this special offer period. And we'll send you another one of your choice–*free!*

It's a one-of-a-kind value your family shouldn't miss.

FISHER-PRICE.® A NAME YOU CAN TRUST.

For years Fisher-Price has been known for helping children develop skills as they play. With Fisher-Price™ Learning Software, your child can develop skills in five key areas: math, language, creativity, basic learning and computer literacy.

Your child will love developing and testing language skills with the help of PETER RABBIT READING, a program that uses voice to teach sound and letter recognition in words.

WINDHAM CLASSICS.™ CLASSIC NOVELS COME TO LIFE.

Pick up Windham Classics, graphic interactive fiction based on some of the greatest adventures of all time. They're great for kids, yet fun for all ages.

Enjoy "THE WIZARD OF OZ.™" You are Dorothy, and each of your decisions determines your adventure. There's more than one adventure to be had as you meet old friends and new in the land of Oz.

WORK BETTER WITH BETTER WORKING.™

Constantly erasing and retyping? Unsure of your spelling? WORD PROCESSOR WITH SPELLCHECKER is exactly what you need. This full featured word processor includes the 50,000 word AMERICAN HERITAGE DICTIONARY™ spellchecker which detects over 99% of misspelled words.

The full line also includes FILE & REPORTand SPREADSHEET–titles that represent the best values on the market today.

SPINNAKER® THE LEADER IN FAMILY LEARNING SOFTWARE.

From pre-school counting to high-school compositions, Spinnaker offers the high quality educational programs you have come to expect. For example, there's the HOMEWORK HELPER™ series which provides step-by-step methods for improving writing and math skills and helps kids tackle otherwise troublesome school assignments.

TELARIUM.™ INTERACTIVE FICTION AT ITS BEST.

Here's graphic interactive fiction by famous writers of science fiction, fantasy, and mystery like Ray Bradbury, Arthur C. Clarke and Erle Stanley Gardner. Each story unfolds with you in the center of the action – and completely in control. In PERRY MASON: THE CASE OF THE MANDARIN MURDER,™ you are the world famous criminal lawyer. Challenge the evidence. Cross-examine the witness. Only you can prove your client's innocence. And time is running out.

A SOFTWARE OFFER FOR ALL AGES.

Any software offer including something for everyone in the family is certainly fantastic. But combine that with a free product offer and what you have is something spectacular. Look for the specially marked boxes of The Family Software Spectacular at your retailer.

The Family Software Spectacular™

TURN PAGE FOR MORE INFORMATION ON THE FAMILY SOFTWARE SPECTACULAR. ⟶

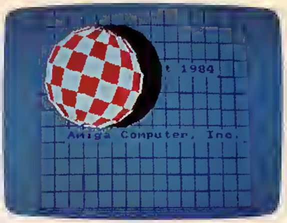
have to type LOAD"$",8 and then LIST—hardly mnemonic or intuitive. But on the Amiga, you can call a directory simply by rolling the mouse to point at a disk icon; the files on the disk will appear onscreen as file folder icons. To delete a disk file, you no longer have to type OPEN15,8,15,"S0:filename-":CLOSE15. Instead, you just point to a file icon and drag it into an icon of a trash can.

With Intuition, you can shrink any screen into a window and layer several such windows on the screen at once. In effect, the computer screen resembles a desktop on which papers can be shuffled around or pushed aside. Windows can be opened, closed, resized, and moved about. You can even display multiple screens on top of each other, all with their own windows.

More advanced users haven't been forgotten, however. Below this shell of windows and menus lies the core operating system, AmigaDOS—perhaps the most powerful disk operating system offered on any personal computer. It's a command-line interpreter patterned after Unix, and it also resembles PC-DOS and CP/M. A large number of advanced functions—including batch files and multitasking DOS commands—are available by typing keyboard commands at the AmigaDOS screen prompt. In fact, AmigaDOS even qualifies as a small programming language. It has commands for IF-THEN comparisons, branching to labels, and looping, so you can construct batch files to run the computer automatically.

Furthermore, AmigaDOS was designed from the ground up as a multitasking operating system. Although it is difficult to pick the Amiga's most impressive feature, multitasking is a top candidate. In effect, it's like having a mainframe computer with several terminals all to yourself. You can run several programs at once, in multiple windows and screens, without noticeably affecting performance.

For instance, you can run a word processor, spreadsheet, and database manager simultaneously, flipping between the three windows as needed. Or you can print out a document with a word processor in one window while writing another document in a different window. Or you can work on several files at once—and even several versions of the same file—by running a single application program in several windows. Programmers can test-run a program in one window while editing the code in another. Even AmigaDOS itself can be running in multiple windows, processing a number of DOS commands simultaneously.

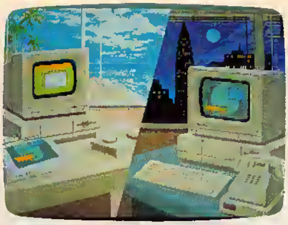


An example of blitter animation. In this demo, the ball spins and bounces around the screen, with sound effects in stereo (see text).

The limit on this kind of multitasking depends on the complexity of the application programs and the amount of available memory. In a test using small BASIC programs, Commodore claims that AmigaDOS has handled 50 windows running 50 programs at once. After that point, they lost track of what was happening.

Part of the secret behind the Amiga's multitasking is its trio of custom chips. Like a team of busy assistants, they free the 68000 microprocessor for more important jobs, sometimes to a startling degree. For instance, a graphics demo on the Amiga features a bouncing ball (see photo). The large checkered ball rotates on its axis in simulated 3-D while bouncing off the bottom and sides of the screen; the shadow it casts is transparent, partially obscuring the background text over which it passes; and bouncing sounds echo realistically from the left and right stereo speakers each time the ball hits a surface. Yet, while all this is happening, the 68000 is doing nothing but calculating the bounce angles, working at only 8 percent capacity.

The blitter and copper are capable of cartoon-quality animation.

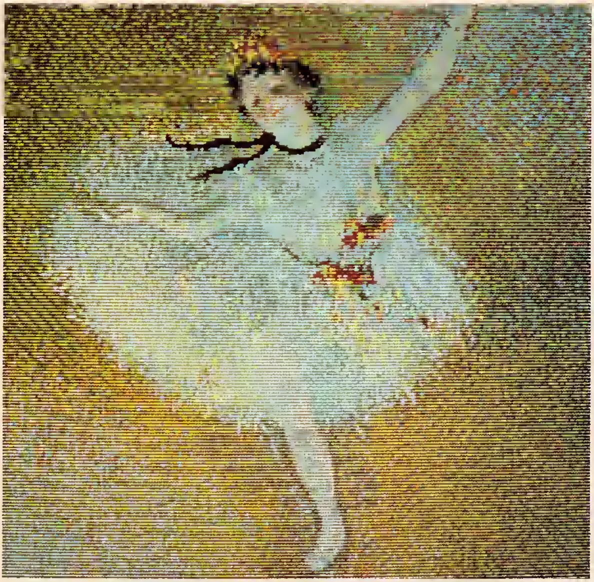


Another low-resolution screen created by Island Graphics. The artist used GraphiCraft, *a drawing program designed by the company that will be sold under the Amiga brand name.*

In fact, blitter animation is so good that Commodore hardly talks about the Amiga's sprite graphics. The blitter can move a screen object of any size, shape, and color at least as fast as a sprite. It even has such sprite-like features as proximity detection and display priorities. One Amiga demo shows a futuristic street scene with moving objects passing behind and in front of each other on five levels—all without sprites. If you do choose to write a program with sprites and use up all eight, the blitter can simulate extra sprites to give you as many independent objects as you want.

Another fascinating feature of the Amiga is its ability to superimpose multiple screens, referred to as *playfields*. You can think of a playfield as a giant sprite that covers the entire screen. By cutting holes in the playfield, you can see the other playfield which lies below it. Each playfield can be independently scrolled vertically and horizontally. In combination with sprites and blitter objects, this feature could lead to incredible 3-D games and other graphics effects. Intuition uses playfields to let you slide one screen away to reveal another beneath it, like a sliding chalkboard.

Even more interesting things become possible when you add an optional video board (about $200). This lets you feed standard video signals into the Amiga and mix them with graphics. The video signals can originate from a video camera, videocassette recorder, laserdisc player, TV receiver with video output, or another computer. Island Graphics of Sausalito, California, which is developing graphics software for the Amiga, used video mixing to reproduce the



This Edgar Degas painting was carefully copied onto the Amiga's low-resolution screen by Island Graphics (see text). Although the 320 × 200 resolution in this mode is no greater than that found on today's home computers, the Amiga's extensive color palette allows it to do more justice to the original.

Degas painting seen in the accompanying screen photo. First, the painting was displayed onscreen as a video image; next, a drawing program was superimposed; then, pixel by pixel, an artist traced the image in computer graphics by manipulating the mouse.

When the optional video board is finished, this process will be automated by a feature called the *frame grabber.* As the term implies, the frame grabber can digitize an incoming video image automatically. You could capture any scene with a video camera, digitize it, modify it with a graphics program if desired, and then dump the image to a graphics printer. The Diablo color inkjet printer, with an Amiga printer driver, can closely reproduce any Amiga screen. We've also heard that work is underway on a laser printer capable of reproducing any screen image in color.

Equally remarkable are the Amiga's sound capabilities. On most computers, four sound channels mean you're limited to four-part harmony or four-note chords. But because the Amiga creates sounds by simulating complex waveforms, it can play chords using only one sound channel. As a result, the Amiga can simulate a wide variety of musical instruments, often with uncanny realism. We've experimented with pipe organ sounds that would grace a cathedral, drum sounds that could hammer out a hot rap rhythm, and heavy-metal electric guitar chords that could blow you out of the room.

The sound demo program we used lets you tinker with the synthesized instruments merely by pulling down menus and selecting options with the mouse. No PEEKs, POKEs, programming skills, or computer knowledge is required. For instance, one menu contained parameters for the sound envelopes, such as attack, decay, sustain, and release. Submenus for each parameter presented such choices as "very slow" to "very fast." By readjusting the electric guitar envelope for a very slow attack and very fast release, we created a backwards guitar sound reminiscent of 1960s records by Jimi Hendrix or the Beatles.

On other computers, custom sounds can only be created by laborious programming. But with an optional accessory (price unannounced), the Amiga provides a shortcut—digital sound sampling. Just as the frame grabber lets you digitize a picture, sampling lets you capture and digitize any sound fed into the Amiga from an outside source. Want to simulate a saxophone? Just play a sax into a sound system that's plugged into the Amiga, or even hook up your stereo to the computer and pipe in some music from a favorite record, tape, or compact disc. We've also heard demos of digitally sampled speech—not to be confused with synthesized speech—that sound as good as tape recordings.

Commodore says several companies are working on music keyboards that will turn the Amiga into a full-blown synthesizer. By using the computer's memory as a sequencer, the Amiga could become a multitrack recording studio for the additional cost of only a few hundred dollars.

This report only scratches the surface. A complete set of technical manuals for the Amiga resembles a stack of Manhattan phone books—it will be months, perhaps years, before they're fully explored by programmers and software manufacturers. People are still developing new techniques on computers which have been available for years, and the Amiga is a whole order of magnitude more advanced.

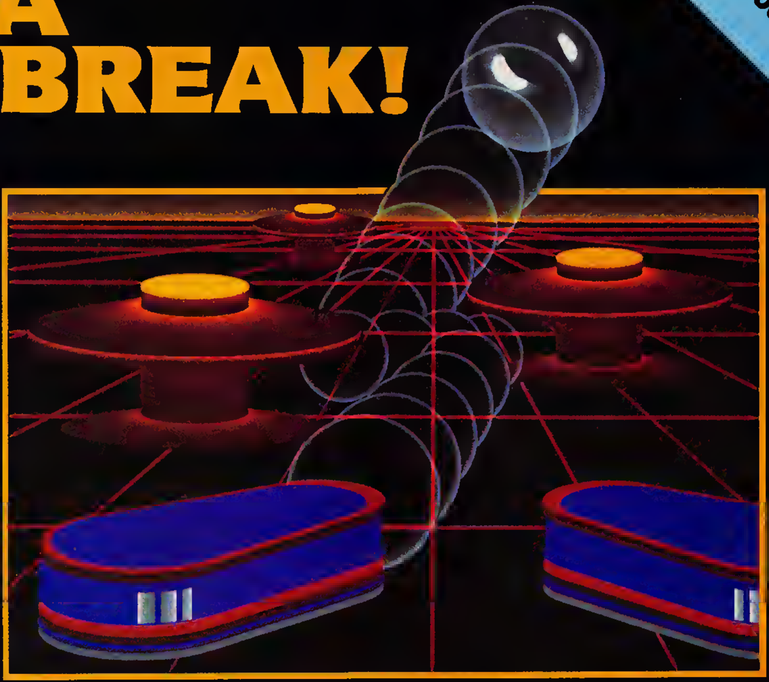
A significant number of companies are now programming for the Amiga, and it appears that about two dozen packages will be available around the time the computer hits the stores. These include everything from word processors to business-graphics programs to games.

Looking toward the future, Commodore says this computer is just the first in a series of Amigas, and that this one represents the low end. What's to follow? Commodore isn't saying. Perhaps the best thing about the Amiga is that it stretches our imaginations a little bit further.

©

TAKE A BREAK!

For Apple, Atari®, & Commodore 64™



WITH NIGHT MISSION

PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

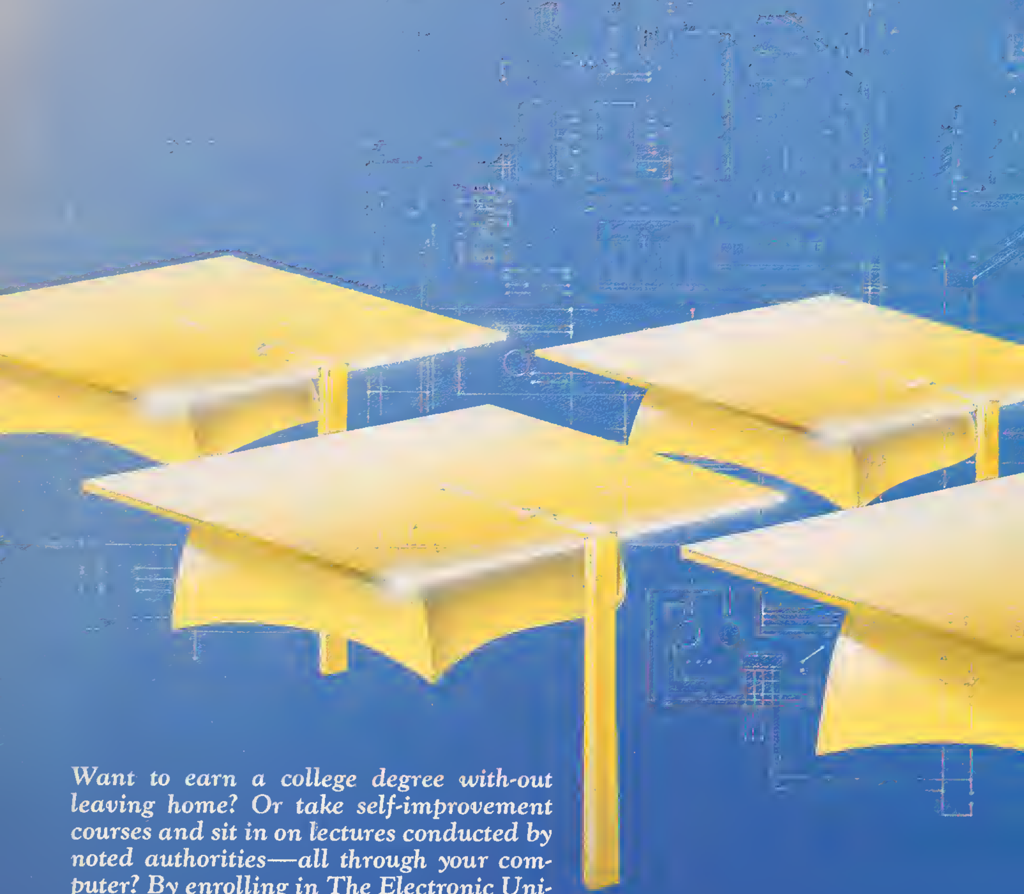
See your dealer . . .

or write or call for more information.

Order Line: 800 / 637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

The Electronic



Want to earn a college degree with-out leaving home? Or take self-improvement courses and sit in on lectures conducted by noted authorities—all through your computer? By enrolling in The Electronic University—a computerized correspondence school—you can do all that, and much more.

University

Sharon Darling

Leeann Pearce calls The Electronic University a "miracle." As she sits at the Commodore SX-64 in her home in West Des Moines, Iowa, Pearce is working toward a degree in computer technology to be granted by Thomas A. Edison College in New Jersey. Although she lives a thousand miles away and suffers from multiple sclerosis, Pearce is gaining the benefits of a college education by using an online educational system designed to work with home computers. Her husband, Frank, is using the same system after he comes home from work at night to earn a master's degree in business. And their eight-year-old daughter, Katie, who used to have trouble with math in school, has boosted her grades by taking an online math tutoring class. Katie is also halfway through a computer programming course and is registering for a class in literary arts this fall. One of the family's biggest problems now is arranging schedules so that each has enough time with the computer.

ble to attending local colleges. But what really made the difference was the ability to take courses without leaving home. Because classes proceed at the student's own pace, Pearce was able to undergo surgery six months ago without interrupting her coursework. And academically, they find the classes as worthwhile as those taken the traditional way.

"I would say the courses are challenging enough," says Pearce. "They're like peanuts—you keep wanting to come back for more. And to bat around ideas with a Ph.D. is really wonderful to me."

What began as a project to teach people how to use modems has grown into a telecommunications network which allows students to use computers to earn high school and college degrees, take noncredit self-improvement courses, and "attend" seminars conducted by noted authorities. Graduate degrees in business administration have even been added to The Electronic University, which was developed by TeleLearning Systems, Inc. of San Francisco, a company founded in 1983 by entrepreneur Ron Gordon.

Close to 15,000 students are now taking classes and seminars in subjects ranging from economics to the subtleties of California wines. And the number of colleges and universities participating in The Electronic University has topped 1,700—all of which offer credit for courses taken through EU. Among the major institutions participating in EU are Cornell University, American University in Washington, D.C., Boston University, Virginia Tech, the New York Institute of Technology, Brigham Young University, the California State University system, the State University of New York, and many other state university systems. If enough coursework is completed to obtain a degree, the diploma is issued by the participating institution, not EU. It's up to students to make sure they meet the requirements of the college from which they want to receive the credit. EU has counseling services, however, to guide students through a degree program.

Close to 15,000 students are now taking classes and seminars in subjects ranging from economics to the subtleties of California wines. And the number of colleges and universities participating in The Electronic University has topped 1,700.

All it takes to enroll in EU is a computer (the system is compatible with the Commodore 64, IBM PC/PCjr, and Apple II series), a modem, and an enrollment package from EU. The package, a one-time investment, costs $79.95 for the Commodore 64 and $149.95 for Apple and IBM computers. If you don't own a modem, TeleLearning will sell you one for about $100.

Tuition ranges from $12 for a seminar up to $295 for some courses leading to a degree. In addition, students pay connect-time fees to participate in seminars and to access the more than 60 online databases. These fees range from about 17 to 80 cents per minute, depending on which database is accessed and when the call is placed. (A $15 monthly minimum is required.) To avoid long-distance charges, the phone calls are made to a local network number.

EU offers seven degree programs, including associate degrees in science, management, and the arts; bachelor's degrees in business administration and the arts; and three master of business administration (MBA) degrees—a general MBA and two specialized MBAs in technology/engineering management and individual financial planning.

Courses for college credit and self-improvement aren't the only services available. The enrollment package also offers tutoring programs for children, an electronic library with more than eight million books, counseling services, and courses in business and professional skills. Once students receive the enrollment package, they can sign up for whatever services they want. Credit courses begin every 60 days.

After students register, they're mailed an information packet on the courses they selected. The packet includes assignment outlines, a list of textbooks and other required materials, and the procedures of the institution delivering the course.

Students also receive a floppy disk containing a general introduction and a series of lessons. A typical lesson might include onscreen instruction, a textbook reading assignment, or other outside activities assigned by the instructor. Periodically, students must use their computer to transmit a progress report to their instructor via electronic mail (E-mail). They can also send questions about the course material and receive answers from the instructor by E-mail. Instructors respond to E-mail messages within 24 hours. In addition, students can schedule an online conference with the professor during designated office hours.

Some courses feature online exchanges with the instructor and even electronic forums with other students—a kind of class discussion via computer. Seminars also employ realtime conferences. Students sign on with their computers at the appropriate time, and the entire discussion session is carried out online.

Roughly 50 percent of a course's contents call for responses from the instructor. A typical class has 10 or 12 lessons; of those, half usually require students to write a response and send it to the instructor via modem, while the other half are "read-write" lessons. In that mode, students read material and type responses on the screen, but the results are not sent to the instructor. However, the instructor has the option of testing students on read-write material to check their progress.

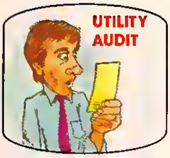
Get the jump on the weatherman by accurately forecasting the local weather yourself!

A scientifically proven way to develop an awesome memory.

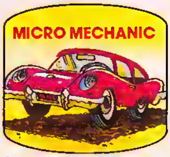
You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

Take control of your personal finances in less than one hour a month.

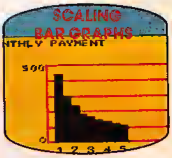
The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!



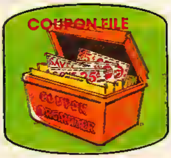
Cut your energy costs by monitoring your phone, electric and gas bills.



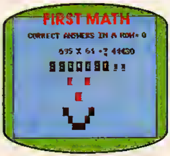
Computerize car maintenance to improve auto performance, economy and resale value.



Create multi-colored bar graphs with a surprisingly small amount of memory.



A time-saving organizer for coupons, receipts and more.



School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.

A real brainflexer. Deflect random balls into targets on a constantly changing playfield.

A fun way to dramatically increase typing speed and accuracy.

Get up to 30 new programs and games for less than 15 cents each—every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand-new, ready-to-run computer programs, even arcade-quality games.

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager, Retirement Planner, Coupon Filer, Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom, and High Speed Mazer.

Your children will find learning fast and fun with First Math, Guess That Animal, and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE!. All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI-99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

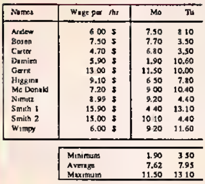
Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

For Faster
Service
Call Toll-Free
1-800-334-0868

COMPUTE! P.O. Box 914, Farmingdale, NY 11737

SUPER PRODUCTIVITY

POWER PLAN

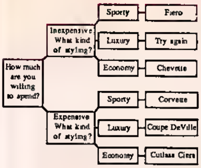


Names	Wage per /hr		Mo	Tu
Ardew	6 00	$	7.50	8 10
Bosen	7.50	$	7.70	3.50
Carter	4.70	$	6.80	3.50
Damien	5.90	$	1.90	10.60
Garrit	13 00	$	11.50	10.00
Higgins	9.10	$	6 50	7.80
Mc Donald	7.20	$	9 00	10.40
Nimitz	8.99	$	9.20	4.40
Smith 1	15.90	$	4 40	13.10
Smith 2	15.00	$	10 10	4.40
Wimpy	6.00	$	9 20	11.60

Minimum	1.90	3 50
Average	7.62	7.95
Maximum	11.50	13 10

Powerful spreadsheet *plus* builtin graphics - display your important data visually as well as numerically. You'll learn fast with the 90+ HELP screens. Advanced users can use the shortcut commands. For complex spreadsheets, you can use POWER PLAN's impressive features: cell formatting, text formatting, cell protection, windowing, math functions, row and column sort, more. Then quickly display your results in graphics format in a variety of 2D and 3D charts. Includes system diskette and user's handbook. **$49.95**

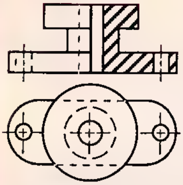
XPER - expert system



How much are you willing to spend?
Inexpensive What kind of styling?
Expensive What kind of styling?
Sporty
Fiero
Luxury
Try again
Economy
Chevette
Sporty
Corvette
Luxury
Coupe DeVille
Economy
Cutlass Ciera

XPER is the first *expert system* - a new breed of intelligent software for the C-64 & C-128. While ordinary data base systems are good at reproducing facts, **XPER** can help you make decisions. Using its simple entry editor, you build the information into a *knowledge base*. XPER's very efficient searching techniques then guide you through even the most complex decision making criteria. Full reporting and data editing. Currently used by doctors, scientists and research professionals. **$59.95**

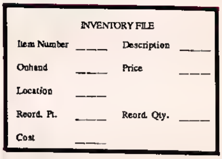
CADPAK Revised Version



CADPAK is a superb design and drawing tool. You can draw directly on the screen from keyboard or using optional lightpen. POINTs, LINEs, BOXes, CIRCLEs, and ELLIPSEs; fill with solids or patterns; free-hand DRAW; ZOOM-in for intricate design of small section. Mesuring and scaling aids. Exact positioning using our *AccuPoint* cursor positioning. Using the powerful **OBJECT EDITOR** you can define new fonts, furniture, circuitry, etc. Hardcopy to most printers. **$39.95**

McPen lightpen, optional **$49.95**

DATAMAT - data management



INVENTORY FILE

Item Number ____ Description ____

Onhand ____ Price ____

Location ____

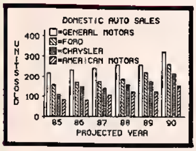
Reord. Pt. ____ Reord. Qty. ____

Cost ____

"Best data base manager under $50"
RUN Magazine

Easy-to-use, yet versatile and powerful features. Clear menus guide you from function to function. Free-form design of data base with up to 50 fields and 2000 records per diskette (space dependent). Simple data base design. Convenient and quick data entry. Full data editing capabilities. Complete reporting: sort on multiple fields and select records for printing in your specific format. **$39.95**

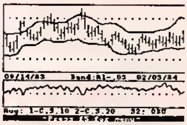
CHARTPAK



Make professional quality charts from your data in minutes. Quickly enter, edit, save and recall your data. Then interactively build pie, bar, line or scatter graph. You can specify scaling, labeling and positioning and watch **CHARTPAK** instantly draw the chart in any of 8 different formats. Change the format immediately and draw another chart. Incudes statistical routines for average, deviation, least squares and forecasting. Hardcopy to most printers. **$39.95**

CHARTPLOT-64 for 1520 plotter **$39.95**

TAS - technical analysis



Technical analysis charting package to help the serious investor. Enter your data at keyboard or capture it through **DJN/RS** or Warner Services. Track high, low, close, volume, bid and ask. Place up to 300 periods of information for 10 different stocks on each data diskette. Build a variety of charts on the split screen combining information from 7 types of moving averages, 3 types of oscillators, trading bands, least squares, 5 different volume indicators, relative charts, much more. Hardcopy to most printers. **$84.95**

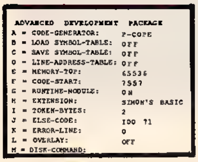
SUPER LANGUAGES

The most advanced C development package available for the C-64 or C-128 with very complete source editor; full K&R compiler (w/o bit fields); linker (binds up to 7 separate modules); and set of disk utilities. Very complete editor handles search/replace, 80 column display with horizontal scrolling and 41K source files. The I/O library supports standard functions like printf and fprintf. Free runtime package included. For C-64/C-128 with 1541/1571 drive. Includes system diskette and user's handbook. **$79.95**

Super Pascal Development Package
Complete Pascal Compiler and Tools for C-64 & C-128

Not just a compiler, but a complete development system. Rivals Turbo Pascal© in both speed and features. Produces fast 6510 machine code. Includes advanced source file editor; full Jensen & Wirth compiler with system programming extensions, new high speed DOS (3 times faster); builtin assembler for specialized requirements. Overlays, 11-digit arithmetic, debugging tools, graphics routines, much more. Free runtime package. Includes system diskette and complete user's handbook. **$59.95**

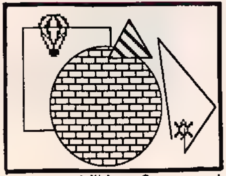
BASIC-64 full compiler



```
ADVANCED  DEVELOPMENT  PACKAGE
A = CODE-GENERATOR:        P-COPE
B = LOAD SYMBOL-TABLE:     OFF
C = SAVE SYMBOL-TABLE:     OFF
O = LINE-ADDRESS-TABLE:    OFF
E = MEMORY-TOP:            65536
F = CODE-START:            7557
G = RUNTIME-NODULE:        ON
H = EXTENSION:             SIMON'S BASIC
I = TOKEN-BYTES:           2
J = ELSE-CODE:             IDO 71
K = ERROR-LINE:            0
L = OVERLAY:               OFF
M = DISK-COMMAND:
```

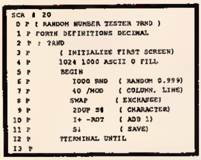
The most advanced **BASIC** compiler available for the C-64. Our bestselling software product. Compiles to superfast 6510 machine code or very compact speed-. code. You can even mix the two in one program. Compiles the complete BASIC language. Flexible memory management and overlay options make it perfect for all program development needs. **BASIC 64** increases the speed of your programs from 3 to 20 times. Free runtime package. Includes system diskette and user's handbook. **$39.95**

VIDEO BASIC development



The most advanced graphics development package available for the C-64. Adds dozens of powerful commands to standard BASIC so that you can use the hidden graphics and sound capabilities. Commands for hires, multicolor, sprite and turtle graphics, simple and complex music and sound, hardcopy to most printers, memory management, more. Used by professional programmers for commerical software development. Free runtime package. Includes system diskette and user's handbook. **$39.95**

FORTH Language



```
SCR # 20
 0 P ( RANDOM NUMBER TESTER ?RND )
 1 P FORTH DEFINITIONS DECIMAL
 2 P : ?RND
 3 P       ( INITIALIZE FIRST SCREEN)
 4 P       1024 1000 ASCII 0 FILL
 5 P       BEGIN
 6 P          1000 RND   ( RANDOM 0.999)
 7 P          40 /MOD    ( COLUMN, LINE)
 8 P         SWAP       ( EXCHANGE)
 9 P          2DUP S@    ( CHARACTER)
10 P          1+ -ROT    ( ADD 1)
11 P          S!         ( SAVE)
12 P       ?TERMINAL UNTIL
13 P
```

Our **FORTH** language is based on the Forth 79 standard, but also includes much of the 83 level to give you 3 times vocabulary of fig-Forth. Includes full-screen editor, complete Forth-style assembler, set of programming tools and numerous sample programs to get you deeply involved in the FORTH language. Our enhanced vocabulary supports both hires and lores graphics and the sound synthesizer. Includes system diskette with sample programs and user's handbook. **$39.95**

Other software also available!
Call now for free catalog and the name of your nearest dealer. Phone: **616/241-5510.**

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 616/241-5510

For fast service call 616/241-5510. For postage and handling, include $4.00 per order. Foreign orders include $8.00 per item. Money orders and checks in U.S. dollars only. Mastercard, Visa and Amex accepted.

Dealer Inquiries Welcome
More than 1200 dealers nationwide

EU does not administer any final exams. There is a practice exam available to students, but it doesn't replace taking a proctored exam at a nearby library or college, which is given by the school accepting the credits. Students also have the option of taking a CLEP (College Level Examination Program) test for credit, or an ACT PEP exam, which is given by the American College Testing Proficiency Examination Program.

Developing a college course to be taught by computer and keeping the material interesting is quite a challenge, says Tom Copley, an EU professor who formerly taught business courses at Antioch College in Ohio. Copley says he was "immediately intrigued" by the idea of an electronic college when he first read about TeleLearning last spring. Not only has he been a computer buff for the last 10 or 15 years, but he also has taken traditional evening school courses in the past. In addition to teaching classes, he's now deeply involved in developing courses for the online school.

"In the first place, you're working with a totally different media, and in order to be effective, you have to take advantage of its advantages. Unfortunately, the cathode ray tube is not nearly as expressive a medium [as books or lectures]." Therefore, he says, "you have to get high learning impact in a small amount of space."

Copley tries to focus on higher-level questions, the kind in which "the student has to synthesize a lot more information and draw more conclusions. I don't find myself using typical textbook jargon—words like *describe, list, differentiate*, etc. I ask for things that require a little more creative thought."

One less obvious advantage to long-distance learning that Copley has discovered is the opportunity to respond to students on a one-to-one basis by E-mail, even though he never sees the student in person. "So often [while teaching in a traditional college], I've had to respond to so many students at once. This is the opposite extreme. Every student gets an individual response, and it's not something off the top of my head, but a thought-out response."

But there are disadvantages, too. "You lose the group dynamics of working in a class environment; some people find that very stimulating. Of course, a lot of educators are critical of the class environment. They say the students are being spoon-fed, entertained. There is none of that in this system. Alternatively, though, there are a lot of things you can do, like screen layout, to make it interesting."

Ron Gordon, founder of The Electronic University.

Today's EU differs from the original focus of the university, which was to offer noncredit courses for personal improvement. After working with the U.S. Department of Education, TeleLearning realized there was an untapped market of people who could benefit from an alternative to traditional colleges.

When TeleLearning first approached universities with the idea of offering courses by computer, many professors were skeptical. Now, however, the school is gaining acceptance nationwide. By next year, founder Ron Gordon hopes to have 50,000 students enrolled. His ultimate goal is for the system to become the largest of its kind in the world, with millions of students.

Tom Copley predicts that alternatives like EU are "the wave of the future." He says the opportunity to take courses on your own time, at your own pace, and at the setting of your choice appeals to certain kinds of students, especially those in remote locations with no colleges nearby.

EU also tends to attract older students than traditional universities. The usual emphasis on undergraduate students who are 18 to 22 years old doesn't always mesh with "people in their 30s who work maybe ten hours a day and may have a family," explains Copley. "Maybe it's been a lifelong dream of theirs to finish college, or maybe their job depends on them finishing a degree. For them, the traditional college life doesn't fit what they need. They're tired after work, or they want the flexibility they can't get from a regular university."

In the future, Copley is convinced The Electronic University will continue growing as more adults find computerized learning accessible, challenging, and rewarding. "So many marketing people focus on baby boomers, and that's where the market is—adults. And that's what undergraduate schools are finding out."

For more information about The Electronic University, contact TeleLearning Systems, Inc., 505 Beach Street, San Francisco, CA 94133. ©

THE #1 BACKUP COPY SYSTEM
IS NOW BETTER THAN EVER!

THE AUTOMATIC DISK BACKUP SYSTEM FOR COMMODORE 64 SOFTWARE

- The LATEST breakthrough in copy technology!
- Copies NEW custom G.C.R. code protection
- Accesses & copies NEW no-sync data blocks!
- Now includes NEW long data-block logic
- Copies normal & PROTECTED data up to track 40
- 3 New IMPROVED copiers! Includes FAST Copier, VARI-SECTOR Copier, and AUTO-LOGIC Nibbler
- Actually copies a FULL disk in only 2.75 minutes!
- Includes one FREE Masterkey Module!
- Inexpensive MODULE releases keep you up-to-date
- MODULES available individually or by subscription
- Works with DiskMaker TOOLKIT and FAT-TRACK Kit
- Requires standard 1541 disk drive or SX-64
- Special version available for INDUS GT drive!
- PRODUCT SUPPORT tel. # & future module updates
- You've tried the rest, now get the BEST!!

THE FASTEST NIBBLE COPIER YOU CAN BUY!

ONLY
$49.95
for copier system

To order CALL **805-687-1541 ext. 94**
Technical line 805-682-4000 ext. 55

MasterCard VISA AMERICAN EXPRESS

Order by phone 24 hrs. 7 days or send cashier's check/money order payable to BASIX. Visa, MasterCard & American Express include card # and exp. date. CA residents add 6% tax. Allow 2 weeks for clearance of personal and co. checks. Add $3.50 shipping/handling for continental U.S. $5.50 for UPS air. COD's shipped to 48 states only, add $4.00. AK, HI, APO, FPO: add $7.50. CANADA add $10.00. Other Foreign orders add $15.00, and remit certified U.S. funds only. No credit card orders from outside North America. Dealers & Distributors invited & supported.

DISKMAKER...THE ONLY COPY SYSTEM YOU'LL EVER NEED!

basix

3463 State Street · Suite 1541F · Santa Barbara · CA 93105

Word Search

Original Program By Michael B. Williams

This computerized puzzle-maker can provide hours of challenging fun. We've included versions for Commodore, IBM PC/PCjr, Apple II-series, TI-99/4A, and Atari computers. A printer is required.

You're probably familiar with word search puzzles: Certain words are hidden in a rectangle of nonsense letters, and it's your job to hunt them down. "Word Search" lets you create such puzzles on your computer's printer with words of your own choice. Since you design the puzzle, you can make it as easy or as difficult as you want, using up to 100 different words on some computers. Topical puzzles make the game even more interesting. For example, you might include only computer words, the names of foreign cities, or stumpers like "uxorious" and "bougainvillaea." Parents and teachers can make puzzles for children using weekly vocabulary lists.

If you're using an Atari, type in and save Program 8, then skip to the program instructions below. For other computers, we've saved space by listing Word Search in the form of one main program with separate line changes and additions for each specific machine. If you're using a Commodore, Apple, IBM PC/PCjr, or TI-99/4A, the first step is to find the specific listing for your computer. Before typing anything, cross out every line in the main program (Program 1) that has the *same* line number as a line in the listing for your computer. Then type in all the lines listed for your computer, as well as all the lines in Program 1 that haven't been crossed out.

No matter which computer you're using, save a copy of Word Search and refer to the notes below before running the program. The following instructions apply to every version:

Word Search begins by asking you for the number of words to be hidden. When you've answered that question, the computer asks you to choose the number of rows and columns for the puzzle grid. Since the grid must be big enough to hide all the words, the computer tells you when you've made the grid too small and lets you try again.

Next, Word Search lets you enter the words one by one. There's no particular limit on word length, but keep in mind that the words must fit inside the grid. (For e ample, you can't fit a 12-letter wor in a 6 × 6 grid.) Since longer words are harder to fit into the grid, the computer sorts the words by length (from longest to shortest) so it can place the longest words first. When many words are involved, this can take a few minutes, so be patient.

Once the words are sorted, you're allowed to name the puzzle. You also have the option of printing the solution to the puzzle (parents and teachers might want to separate the solution from the puzzle until the puzzle has been tried). After printing one puzzle, you can create another, using the same word list (the words will be rearranged) or entirely new words. Word Search is designed to permit a maximum of 100 words in a 99 × 99 grid (exceptions for certain computers are noted below). However, puzzles of that size can take a long time to create—over an hour in some cases. In addition, many

"Plinkers" Just Want To Have Fun.

Every night, American PEOPLE/LINK subscribers (Plinkers) go online and party with other men and women of all ages and interests from across the country and the world.

What do they do? They do what adults do. They make dates and play trivia, they tell jokes and argue, they form clubs, and even live out their fantasies. But the one thing they all have in common is that they want to have *fun*.

Our subscribers develop friendships and romances as they "talk live" (in groups or in private) and send electronic mail to each other. And our low rates will show you why Plinkers can *afford* to have so much fun.

To discover our affordable videotex entertainment, call or write us today.

PEOPLE/LINK . . . for the *fun* of it.

AMERICAN
PEOPLE
LINK

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road – Suite 1505
Arlington Heights, IL 60004

1-800-524-0100 (Toll Free) 1-312-870-5200 (Illinois)

VISA MasterCard

USE YOUR VISA, MASTERCARD
OR AMERICAN EXPRESS.

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- ✓ THE BEST PRICES!
- ✓ Next day shipping on all in stock items.
- ✓ Free easy access order inquiry.
- ✓ Orders from outside Pennsylvania and Nevada save state sales tax.
- ✓ Free technicial support with our factory trained technical staff.
- ✓ There is no limit and no deposit on C.O.D. orders.
- ✓ There's no extra charge for using your credit card. Your card is not charged until we ship.
- ✓ No waiting period for cashiers checks.
- ✓ We accept purchase orders from qualified corporations. Subject to approval.
- ✓ Educational discounts available to qualified institutions.
- ✓ FREE CATALOG MEMBERSHIP.

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215

CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A209

MAILING ADDRESS

Computer Mail Order
Dept. A209
477 East Third Street
Williamsport, PA 17701

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard AMERICAN EXPRESS DINERS CLUB

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

ATARI

COMPUTER CARTRIDGES

$2.99

AAB4013 ASTEROIDS
AAB8130 CAVERNS OF MARS
AAB4020 CENTIPEDE
AAB4025 DEFENDER
AAB8026 DIG DUG
AAB8031 DONKEY KONG
AAB8039 EASTERN FRONT
AAB8030 E.T.
AAB4024 GALAXIAN
AAB4012 MISSILE COMMAND
AAB4022 PAC MAN
AAB4027 OIX
AAB4008 SPACE INVADERS
AAB4011 STAR RAIDERS
AAB4008 SUPER BREAKOUT
AAB4010 3-D TIC-TAC-TOE
AAB8045 MUSIC COMPOSER
AAB8045 PENGO
AAB8029 FOOTBALL
AAB8057 TYPO ATTACK

(Repeckaged without instructions)

COMPUTERS

Atari 400	**$39.99**
Atari 800 48K	**$69.99**
Atari 1200 64K	**$69.99**

COMPUTER PACKAGES

800, 810, 410, Basic, Disks, DC	**$239.00**
400, 410, Basic, DC	**$59.99**
1200, 810, 825, 850	**$379.00**
400, 20-cart, 482 Educator 822	**$119.00**

(Warranty packages available)

ACCESSORIES

Flip & File + 8 Cartridges	$34.99
Ateri 850 Interface Module	$99.99
Atari 835 Direct Modem	$39.99
Ateri 410 Program Recorder	$19.99
Atari 822 Thermel Printer	$19.99
Atari 482 The Educetor	$24.99
Atari 483 The Progremmer	$19.99
Ateri Basic	$14.99
Ateri Writer	$14.99
Ateri AAB4018 Pilot	$9.99
Atari AAB7097 Logo	$9.99

HOME COMPUTERS

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

Macintosh Software

Letus Jazz	CALL
Microsoft Excel	CALL
Microsoft Business Pak	$375.00
Living Videotaxt ThinkTank 512	$159.00
Manhetten Ready, Set, Go	$79.99
Creighton Davelopment Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Beck to Basics - GL	$109.00
PFS File & Report (New Version)	$129.00
Silicen Beach Airborn.	$25.99

commodore

C128 Computer	**$NEW**
C1571 (Disk Drive for C128)	**$NEW**
C1902 (RGB 13" Monitor for C128)	**$NEW**
C1670 (Modem for C128)	**$NEW**
SX-64 Portable	CALL
Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$189 00
M-802 Dot Matrix/Serial	$219.00
MCS 803 Dot Matrix	$179 00
C1802 Color Monitor	$199 00
C1660 Auto Modem	$59.99
DPS 1101 Deisy Printer	$339 00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 71B	$419 99
HP 11C	$62 99
HP 12C/15C/16C	$89.99
HP 75D	$999 99
HPIL Module	$98 99
HPIL Cassette or Printer	$359 99
Card Reader	$143.99
Extended Function Module	$63 99
Time Module	$63 99

We steck the full lina of HP celculetor products

NEC

PC-8401	CALL
PC-8201 Portable Computer	$289.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159 99
PC-1260	$109.99
PC-1500A	$165 99
PC-1250A	$88.99
CE-125 Printer/Cassette	$128 99
CE-150 Color Printer Cassette	$171.99
CE-161 16K RAM	$134.99

DISKETTES

maxell.

3½" SS/DD	$39.99
3½" DS/DD	$54.99
5¼" MD-1 w/Hardcase	$17.99
5¼" MD-2 w/Hardcase	$23.99
5¼" MD-2-HD for AT	$44 99

Verbatim.

5¼" SS/DD	$21.99
5¼" DS/DD	$29.99
Disk Analyzer	$24 99

Dennison

Elephent 5¼" SS/SD	$13.99
Elephent 5¼" SS/DD	$15.99
Elephant 5¼" DS/DD	$16.99
Elephant Premium DS/DD	$23.99

IBM

5¼" DS/DD Floppy Disks (Box of 10)	$25.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-in-File 10	$3.99
Flip-in-File 50	$17.99
Flip-in-File 50 w/lock	$24.99
Flip-in-File (400/800 ROM)	$11.99

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$8.99

SMITH & BELLOWS

Wood Disk Holders	CALL

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$189.99
Mark XII (1200 Baud)	$259.00
Signalmen Express	$299.00
Lightning 2400 Baud	$399.00

Hayes

Smartmodem 300	$145.00
Smartmodem 1200	$389.00
Smertmodem 1200B	$359.00
Smartmodem 2400	$699.00
Micromodem IIe	$249.00
Smert Com II	$89.99
Chronograph	$199.00
Transet 1000	CALL

AST

Reach 1200 Baud Half Card	$399.00

mpp MICROBITS

MPP-1000E AD/AA (Atari)	$79.99
MPP-1064 AD/AA (C-84)	$69.99

Novation

Smart Cet Plus	$319.00
J-Cat	$99.99
Novation 2400	CALL
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrede	$229.00
Mecmodem	$319.00

QUADRAM

Ouadmodem II

300/1200	$339.00
300/1200/2400	$499.00

TELELEARNING

C64 300 Baud......(Closeout)	$39.99

GRAPHICS

Koala

IBM	$89.99
Apple/Franklin	$79.99

Polaroid

Palette	$1299.00

DRIVES

HARD

ALLOY

PC Stor.	CALL

IDEAssociates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249 00
15 meg 5 Removable/10 Fixed	$2149 00
25 meg 5 Removable/20 Fixed	$2499.00

I-MEGA

10 Meg Bernoulli Box	$2099.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1599 00

TALLGRASS TECHNOLOGIES

COMMITTED TO MEMORY

12, 25, 35, 50, 80 meg (PC)	from $1499 00

FLOPPY

INDUS

Apple GT	$209.00
Atari GT	$239.00
C-64 GT	$259.00

μ-SCI

A1.5 Apple	$199.00
A2 Apple	$199.00

MSD

SD1 C-84 Single	$219.00
SD2 C-64 Dual	$469.00

Tandon

320K 5¼" (PC)	$119.00

TEAC

320K 5¼	$119.00

CALL TOLL-FREE

MONITORS

AMDEK

300 Green	$129.00
300 Amber	$139.00
310 Amber IBM-Plug	$169.00
Color 300 Composite	$239.00
Color 500 Composite/RGB	$389.00
Color 800 Hi-Res (640x240)	$399.00
Color 700 Hi-Res (720x240)	$499.00
Color 710 Long Phosphor	$579.00

NAP

12" Amber/Green Composite	$99.99
12" Amber/Green TTL (ea.)	$119.00

NEC

JB 1260 Green	$59.99
JB 1201/1205 (ea.)	$99.99
JB 1270 Green	$139.00
JB 1275 Amber	$149.00
JB 1280 G TTL/1285 A TTL	$149.00
JC 1410 RGB	$889.00

PRINCETON

MAX-12E Amber	$189.00
HX-9 9" RGB	$469.00
HX-9E Enhanced	$519.00
HX-12 12" RGB	$469.00
HX-12E Enhanced	$559.00
SR-12 Hi-Res	$599.00
SR-12 P Enhanced	$649.00

TAXAN

115 12" Green Mono	$119.00
116 12" Amber Mono	$119.00
121 Green TTL	$139.00
122 Amber TTL	$149.00
420 Hi-Res RGB (IBM)	$429.00
440 Ultra Hi-Res RGB	$559.00

QUADRAM

8400 Quedchrome I	$479.00
8410 Quachrome II	$429.00
8420 Amberchrome	$179.00
8500 Quad Screen	$1499.00

ZENITH

ZVM 122/123	$79.99
ZVM 124 IBM Amber	$149.00
ZVM 130 Color	$269.00
ZVM 131 Color	$249.00
ZVM 133 RGB	$429.00
ZVM 135 RGB/Color	$459.00
ZVM 136 RGB/Color	$599.00
1220, 1230, 1240	CALL

INTERFACES

PRACTICAL PERIPHERALS

Graphcerd	$79.99
Seriall Card	$99.99
Microbuffer II +	$189.00
Microbuffer 32K	$189.00

QUADRAM

Microfazer	from $139.00
Efazer (Epson)	from $79.99

Orange Micro

Greppler CD (C64)	$99.99
Grappler + (Apple)	$89.99
Grappler 16K + (Apple)	$159.00

DIGITAL DEVICES

Ape Face (Atari)	$49.99
U-Print A (Atari)	$54.99
U-A16/Buffer (Atari)	$74.99
U-Call Interface (Atari)	$39.99
U-Print C (C64)	$49.99
P-16 Print Buffer	$74.99

MPP MICROBITS

MPP-1150 Perallel (Atari)	$69.99
MP-1150XL (Atari 1200XL)	$69.99
MicroStuffer 64K Print Buffer	$109.00

PRINTERS

AXIOM

AT-100 Atari Interface Printer	$139.00
AT-550 Atari Dual Mode	$249.00
GP-100 Parallel Interface	$189.00
GP-700 Color Printer	$449.00
GP-550 Perallel Printer	$239.00
Elite 5 Letter Quality	$239.00
Elite 5 C64 Interface	$249.00
Penman 3-pen	$289.00

CITIZEN

MSP-10 (80 col.)	$349.00
MSP-15 (132 col.)	$499.00
MSP-20 (80 col.)	$489.00
MSP-25 (132 col.)	$679.00

C. ITOH

Prowriter 7500	$219.00
Prowriter 8510P	$299.00
Prowriter 8510 NLQ	$329.00
Prowriter 1550P	$469.00
F10-40P Sterwriter	$869.00
F10-55 Printmaster	$1049.00
Prowriter 8510-NLQ	$329.00

corona

Lazer LP-300	$2799.00

DIABLO

D25 Daisywheel	$599.00
630-109 Daisywheel	$1749.00
D801F Daisywheel	CALL

daisywriter

2000	$749.00

EPSON

LX-80, LX-90, FX-85, FX-185 JX-80, DX-10, DX-20, SQ 2000 HS-80, Homewriter 10	CALL

JUKI

6000 Letter Quelity	CALL
8100 Letter Quellty	CALL
8300 Letter Quelity	CALL

NEC

8027 Trensportable	$299.00
2000 Series	$699.00
3000 Series	$1099.00
8000 Series	$1449.00
ELF 360	$449.00

OKIDATA

84, 182, 192, 193, 2410	CALL
Qkimate 10 (Specify C64/Ateri)	$189.00
Qkimate 20 (IBM)	CALL

OLYMPIA

Needlepoint Dot Metrix	$299.00
Compact RQ	$339.00
Compact 2	$369.00

Panasonic

KX1091	$259.00
KX1092	$389.00
KX1093	$479.00

QUADRAM

Quadjet	$399.00

SILVER-REED

500 Letter Quality	$279.00
550 Letter Quelity	$419.00
770 Letter Quality	$759.00

star

SG-10C (C64 Interface)	$NEW
SB/SD/SG/SR Series	CALL
Powertype Letter Quality	CALL

TOSHIBA

1340 (80 column)	$599.00
P351 (132 column)	$1299.00

PC COMPATIBLES

IBM PC SYSTEMS

Configured to your specification.
Call for Best Price!
IBM-PC, IBM-PC II, IBM-XT, IBM-AT

SOFTWARE FOR IBM

ALPHA

Electric Desk IBM-PC	$199.00

ASHTON-TATE

Fremework	$369.00
dBese III	$369.00

BORLAND INTERNATIONAL

Turbo Pascal 3.0	$49.99
Sidekick	$39.99

CENTRAL POINT

Copy II PC-Beckup	$29.99

DECISION RESOURCES

Chartmaster	$259.00
Signmester	$179.00

FOX & OELLER

Quickcode III	$169.00

FUNK SOFTWARE

Sideweys	$39.99

Harvard Software Inc.

Harvard Project Maneger	$209.00
Totel Project Menager	$269.00

Human Edge™

Communication Edge	$99.99
Menegement Edge	$119.00
Negotielion Edge	$139.00
Sales Edge	$119.00

LIFETREE

Volkswriter Duluxe	$159.00

LIVING VIDEOTEXT

Think Tenk	$109.00

Lotus

Symphony	$439.00
1-2-3	$309.00

MECA SOFTWARE

Meneging Your Money	$109.00

MICROSTUF

Crosstalk XVI	$89.99
Microstuf Remote	$89.99

MICRORIM

R:Base 4000	$259.00
R:Base 5000	$399.00
Cloul 2.0	$129.00

MicroPro

WordStar 2000	$249.00
WordStar 2000 +	$309.00
WordStar Professional	$299.00

MICROSOFT.

Word	$239.00
Mouse	$139.00
Flight Simulator	$39.99
MultiPlan	$129.00

MultiMate

Multi Mate Word Proc.	$249.00

NOUMENON

Intuil	$69.99

NORTON

Norton Utilities 3.0	$59.99

Peachtree Software

Peachtext 5000	$169.00
PeachPack (GL/AP/AR)	$219.00

pfs: IBM/APPLE

Access	$54.99
Write/Greph/File/Plan (ea)	$79.99
Report	$74.99
Proof	$59.99

Professional Software

Wordplus-PC w/Boss	$249.00

ROSESOFT

Prokey	$89.99

TNE SOFTWARE GROUP

Enable	$339.00

SATELLITE SYSTEMS

Word Perfect 4.0	$219.00

SORCIM / IUS

Accounting AP/AR/GL/INV/OE (ea.)	$295.00
SuperCelc III	$195.00
EesyWriter II System	$195.00
Super Project	$195.00

SPI

Qpen Access	$379.00

TNOUGNTWARE

Trigger	$289.00
Sell, Sell, Sell Training	$299.00
Sell, Sell, Sell Application	$179.00

AT&T

Safari (7300)	CALL
6300	CALL

corona

PPC400 Dual Portable	$1799.00
PPCXT 10 meg Porteble	$2399.00
PC40022 10 meg Desktop	$1999.00

ITT

ITT X-TRA 256K, 2 Drive System	CALL
ITT X-TRA 256K, 10 meg Hard Drive System	CALL

SANYO

MBC 550-2 Single Drive	$699.00
MBC 555-2 Dual Drive	$969.00
MBC 775 Portable	$1599.00
MBC 511 10 meg	CALL
MBC 675 Porteble	CALL
MBC 880 Desktop	CALL

ZENITH

PC-151-21 Single Desktop	CALL
PC-151-52 Dual Desktop	CALL
PC-151-53 10 meg Desktop	CALL
PC-161-21 Single Portable	CALL
PC-181-52 Dual Portable	CALL
Z-200 (AT)	CALL
171 (Portable)	CALL
138 (Trensportable)	CALL
148 (DeskTop)	CALL

(NEW)

MULTIFUNCTION CARDS

AST

Six Pack Plus	$239.00
Mega Plus II	$269.00
I/Q Plus II	$139.00
Advantage-AT	$399.00
Greph Pak	$599.00
Monogreph Plus	$399.00
Preview Mono	$299.00
PC Net Cards	$379.00
5251/11 On-line	$799.00
5251/12 Remote	$579.00
3780 Emulation Card	$639.00
BSC Bisync	$489.00

dca

IRMA 3270	$879.00
IRMA Print	$999.00

EVEREX

Color Card (Graphics Edge)	$299.00
Megic Cerd	$199.00

HERCULES

Graphics	$299.00
Color	$159.00

IDEAssociates

IDEAmax - ZPR, 64K, C, S, P	$229.00
IDEAmini - YPR, C, S, P	$189.00
IDEAminimax - MPR 128K	$229.00
IDEAshare Software	$219.00
IDEA 5251	$699.00

MYLEX

The Cheirmen	$489.00

PARADISE

Modular Graphics Card	$274.00
Multi Display Card	$289.00
Flve Pack C, S	$159.00

PERSYST

Bob Board	$389.00

PLANTRONICS

Color Plus	$369.00

TECMAR

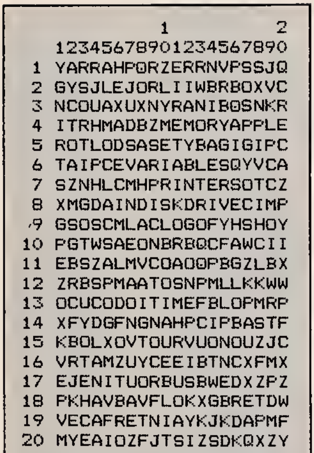
Ceptein - 64	$239.00
Captain Jr. 128K	$339.00
Graphics Master	$469.00

QUADRAM

Quedport-AT	$119.00
Quadmeg - AT (128K)	$349.00
The Gold Quedboard	$449.00
The Silver Quadboerd	$239.00
Expanded Quadboard	$219.00
Quad 512 +	$229.00
Liberty	$309.00
QuedSprint	$499.00
QuadLink	$399.00
Quadcolor 1	$199.00
QuadJr. Expansion Chessis	$469.00
Expansion Chassis Memory	$199.00
Quadmem. Jr.	$199.00
Chronagreph	$79.99
Parallel Intertace Board	$64.99

COMPUTER MAIL ORDER

1-800-233-8950



```
               1         2
    12345678901234567890
  1 YARRAHPORZERRNVPSSJQ
  2 GYSJLEJORLIIWBRBOXVC
  3 NCOUAXUXNYRANIBOSNKR
  4 ITRHMADBZMEMORYAPPLE
  5 ROTLODSASETYBAGIGIPC
  6 TAIPCEVARIABLESQYVCA
  7 SZNHLCMHPRINTERSOTCZ
  8 XMGDAINDISKDRIVECIMP
  9 GSOSCMLACLOGOFYHSHOY
 10 PGTWSAEONBRBQCFAWCII
 11 EBSZALMVCOAOOPBGZLBX
 12 ZRBSPMAATOSNPMLLKKWW
 13 OCUCODOITIMEFBLOPMRP
 14 XFYDGFNGNAHPCIPBASTF
 15 KBOLXOVTOURVUONOUZJC
 16 VRTAMZUYCEEIBTNCXFMX
 17 EJENITUORBUSBWEDXZPZ
 18 PKHAVBAVFLOKXGBRETDW
 19 VECAFRETNIAYKJKDAPMF
 20 MYEAIOZFJTSIZSDKQXZY
```

"Word Search" prints out challenging hidden-word puzzles of various sizes on your printer.

printers can't print more than 80 columns unless you first send the printer a special escape code for condensed type (see your printer manual).

Commodore Versions

The line changes listed as Program 2 are for the Commodore 64, 128, Plus/4, 16, PET, and VIC-20 (with at least 8K expansion). If you're using a VIC with only 8K expansion, type in the line changes shown in Program 2 and also substitute lines 95 and 100 in Program 4. If you're using a Commodore 16, type the line changes from Program 2 and also substitute lines 95 and 100 in Program 3. The VIC with only 8K expansion can hide a maximum of 50 words in a 50 × 50 grid; the 16 is limited to a maximum of 60 words in a 60 × 60 grid. If you're using a PET, you'll have to make similar adjustments, depending on the amount of memory available.

Apple And IBM

The Apple version of Word Search runs on any Apple II-series computer with either DOS 3.3 or ProDOS. Follow the general instructions above, typing in the line changes listed as Program 5. IBM users should enter the line changes in Program 6; this version runs on a PC or PCjr with any memory configuration.

TI Word Search

Program 7 lists the line changes required for TI. The unexpanded TI-99/4A is limited to 50 words in a 50 × 50 grid. However, with memory expansion this number can be increased by changing the value of MC in line 95 from 50 to the desired value. You will also need to increase every occurrence of 50 in line 100 to the same value. Adjust line 2000 for whatever configuration your particular printer requires.

Atari Version

The Atari version of Word Search is complete in itself. Simply type in Program 8, save a copy, and run it. Ataris with 32K or 48K memory can create puzzles with up to 100 words in a 99 × 99 grid. If your Atari has 16K, you're limited to 25 words in a 25 × 25 grid. To run Word Search on a 16K Atari you must make two additional changes in line 100 of Program 8: Change the 99 and the 100 to 25.

Program 1: Word Search (Main Program)

Version By Patrick Parrish, Programming Supervisor

Please refer to the article instructions before entering this listing.

```
95 MC=99
100 DIM FF$(100),S$(99),W$(100
    ),CC(100),RR(100),L(100),E
    $(2,2)
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 READ E$(I+1,J+1)
140 NEXT J
150 NEXT I
160 DATA "NW"," N","NE"," W","
    {2 SPACES}"," E","SW"," S"
    ,"SE"
170 FOR I=1 TO MC
180 G$=G$+" "
190 NEXT I
200 FOR I=1 TO 8
210 READ D(1,I),D(2,I)
220 NEXT I
230 DATA -1,-1,-1,Ø,-1,1,Ø,-1
240 DATA Ø,1,1,-1,1,Ø,1,1
250 GOTO 1220
260 REM SHELL SORT
270 PRINT "SORTING..."
280 X=1
290 X=2*X
300 IF X<=WØ THEN 290
310 X=INT(X/2)
320 IF X<>Ø THEN 340
330 RETURN
340 FOR Y=1 TO WØ-X
350 Z=Y
360 A=Z+X
370 IF L(Z)>=L(A) THEN 460
380 X$=W$(Z)
390 W$(Z)=W$(A)
400 W$(A)=X$
410 B=L(Z)
420 L(Z)=L(A)
430 L(A)=B
440 Z=Z-X
450 IF Z>Ø THEN 360
460 NEXT Y
470 GOTO 310
480 REM HIDE WORDS
490 FOR X=1 TO WØ
500 FOR Y=1 TO 50
510 R1=INT(RND(1)*RØ)
520 C1=INT(RND(1)*CØ)
530 D1=INT(RND(1)*8)+1
540 O1=D1
550 DX=D(1,D1)
560 DY=D(2,D1)
570 IF R1+DX*L(X)<1 OR R1+DX*L
    (X)>RØ OR C1+DY*L(X)<1 THE
    N 590
580 IF C1+DY*L(X)<=CØ THEN 630
590 D1=D1*(D1<8)*(1=1)+1
600 IF D1<>O1 THEN 550
610 NEXT Y
620 GOTO 800
630 FOR Z=1 TO L(X)
640 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 680
650 R1=R1+DX
660 C1=C1+DY
670 IF MID$(S$(R1),C1,1)<>" "
    {SPACE}AND MID$(S$(R1),C1,
    1)<>MID$(W$(X),Z,1) THEN 5
    9Ø
680 NEXT Z
690 FOR Z=L(X) TO 1 STEP -1
700 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 770
710 S$(R1)=MID$(S$(R1),1,C1-1)
    +MID$(W$(X),Z,1)+MID$(S$(R
    1),C1+1)
720 RR(X)=R1
730 CC(X)=C1
740 FF$(X)=E$(DX+1,DY+1)
750 R1=R1-DX
760 C1=C1-DY
770 NEXT Z
780 NEXT X
790 GOTO 890
800 GOSUB 1720
810 PRINT "SORRY, BUT I CAN'T
    {SPACE}FIT WORD NUMBER ";S
    TR$(X);" , ";W$(X);" , ";
820 PRINT "INTO THE GRID. SHOU
    LD I SKIP IT, START OVER,
    {SPACE}OR TRY AGAIN"
830 INPUT X$
840 IF MID$(X$,1,2)="ST" THEN
    {SPACE}1660
850 IF MID$(X$,1,2)="TR" THEN
    {SPACE}500
860 IF MID$(X$,1,2)<>"SK" THEN
     830
870 W$(X)="/"
880 GOTO 780
890 FOR X=1 TO RØ
900 FOR Y=1 TO CØ
910 IF MID$(S$(X),Y,1)<>" " TH
    EN 930
920 S$(X)=MID$(S$(X),1,Y-1)+CH
    R$(INT(26*RND(1)+65))+MID$
    (S$(X),Y+1)
930 NEXT Y
940 NEXT X
950 REM DONE
960 PRINT
970 PRINT "I AM FINISHED. WHAT
     DO YOU WANT TO CALL THE W
    ORD SEARCH"
980 INPUT T$
990 SL=Ø
1000 PRINT
1010 PRINT "DO YOU WANT TO PRI
     NT THE SOLUTION (Y/N)"
1020 GOSUB 1180
1030 IF A$="N" THEN 1050
1040 SL=1
```

The complete 1541 enhancement system . . .

STARDOS

MAKES GREAT COFFEE!!

The 1541 is the slowest disk drive on planet earth. Even simple operations seem to take forever. Quickloaders and Fastloaders that software-patch the operating system are vulnerable to being knocked out of memory, rendering them totally useless. Even Flashier products that require permanent modifications to the 64 and 1541 can't compete with the blinding speed of STARDOS.

STARDOS accelerates every (yes, we said every) function of the 1541 disk drive. Other fast loaders only load PRG files faster. STARDOS also speeds up SEQ, REL, USR and DIRECT ACCESS files. Everything including FORMAT, VERIFY, SCRATCH, VALIDATE, INITIALIZE and COPY are much faster. In addition STARDOS adds a vast array of easy to use commands and utilities all at the touch of a key.

A sampling of STARDOS features:

- Accesses ALL types of files up to 1,000% faster!
- Saves up to 300% faster than normal (with extended verify)
- DOES NOT CHANGE THE SPEED AT WHICH THE DRIVE MOTOR SPINS
- Makes your 1541 MORE RELIABLE and LESS PRONE TO BREAK DOWN or OVERHEAT!
- STARDOS is fully expandable for multiple fast disk drives
- Easy (5 minute) plug in installation. User friendly manual
- 100% compatible with software and serial bus peripherals
- Adds years of life to your disks and drive in reduced wear
- Cures a number of bugs in the Commodore 64 and 1541 disk drive including :
  - The damaging 'Head Knock' that can mis-align your 1541
  - The@: Save with replace bug!!
  - The Editor lock-up bug
- Harness the full power of your disk drive with the built in DOS wedge
- Lock/unlock files and protect/unprotect disks from the keyboard
- Powerful sector editor allows direct viewing of diskette
- Upgrades computer and drive to the latest Commodore specifications
- Instant access to the built in mini-wordprocessor for short notes and memos
- Built in copier copies all file types (even relative) easily and effectively
- Built in disk duplicator copies an entire diskettte in less than 3 minutes
- Fully expanded machine language monitor, always on line
- Qualify hardware. Full 120 day repair or replace warranty
- Satisfaction guaranteed or money back—no questions asked
- (Oh, by the way, we lied, STARDOS makes <u>LOUSY COFFEE</u>)

Simply the best. $74.95
[Take advantage of our introduction special at $64.95]

Personalize your C64! for a $10 (non-refundable) charge we will include a 21 character power-on message with your favorite screen, border and text colors. *Example:*

M	I	K	E	'	S		C	O	M	M	O	D	O	R	E	-	6	4		

Background *Black* Border *Red* Text *Green*

Write of phone for additional information.

*Trademark of Commodore Business Machines

. . . WRITE OR PHONE . . .

STARPOINT SOFTWARE

Star Route 10 Gazelle, CA 96034 [916] 435-2371

When ordering by mail:

- $64.95 + 3.00 shipping
- $64.95 + 4.00 COD orders
- Shipping out of USA $6.00
- Calif. residents add 6% sales tax
- VISA or Mastercard accepted

Please allow 4-6 weeks for delivery.

```
1050 GOSUB 2000
1060 GOSUB 1720
1070 F=0
1080 PRINT "DO YOU WANT ANOTHE
     R GRID (Y/N)"
1090 GOSUB 1180
1100 IF A$="Y" THEN 1120
1110 END
1120 PRINT
1130 PRINT "DO YOU WANT TO USE
      THE SAME WORDS (Y/N)"
1140 GOSUB 1180
1150 IF A$="N" THEN 1280
1160 F=1
1170 GOTO 1340
1180 INPUT A$
1190 IF A$<>"Y" AND A$<>"N" TH
     EN 1180
1200 RETURN
1210 REM INITIALIZATION
1220 GOSUB 1720
1230 LL=6
1240 GOSUB 1740
1250 PRINT "{8 SPACES}WORD SEA
     RCH"
1260 LL=4
1270 GOSUB 1740
1280 FOR I=1 TO W0
1290 W$(I)=""
1300 L(I)=0
1310 NEXT I
1320 PRINT "HOW MANY WORDS WOU
     LD YOU LIKE IN YOUR WORD
     {SPACE}SEARCH"
1330 INPUT W0
1340 PRINT
1350 PRINT "HOW MANY ROWS AND
     {SPACE}COLUMNS IN THE GRI
     D"
1360 INPUT R0,C0
1370 PRINT
1380 PRINT
1390 IF R0*C0>=10*W0 THEN 1440
1400 PRINT "I DON'T THINK I CO
     ULD DO THIS."
1410 FOR I=1 TO 1000
1420 NEXT I
1430 GOTO 1340
1440 PRINT "I THINK I CAN DO T
     HIS."
1450 IF C0<=MC THEN 1470
1460 PRINT "(BUT IT WON'T FIT
     {SPACE}ON THE PAPER.)"
1470 IF F=1 THEN 1660
1480 LL=3
1490 GOSUB 1740
1500 PRINT "ENTER THE ";STR$(W
     0);" WORDS. TO CORRECT A
     {SPACE}MISTAKE, ENTER X"
1510 PRINT
1520 FOR I=1 TO W0
1530 PRINT "WORD NUMBER ";I;":
     "
1540 INPUT X$
1550 IF LEN(X$)<=R0 AND LEN(X$
     )<=C0 AND X$<>"X" THEN 16
     10
1560 IF X$<>"X" THEN 1590
1570 I=I-(I>1)*(1=1)
1580 GOTO 1530
1590 PRINT "OOPS...THE WORD IS
      TOO LONG."
1600 GOTO 1530
1610 W$(I)=X$
1620 L(I)=LEN(X$)
1630 NEXT I
1640 GOSUB 1720
1650 GOSUB 270
1660 PRINT
1670 PRINT "OKAY, I WILL GO TO
      WORK (WISH ME LUCK...)."
1680 FOR I=1 TO R0
1690 S$(I)=LEFT$(G$,C0)
1700 NEXT I
1710 GOTO 490
1730 RETURN
1740 FOR I=1 TO LL
1750 PRINT
1760 NEXT I
1770 RETURN
1999 REM PRINTER ROUTINE
```

Program 2: Line Changes For Commodore 64, 128, Plus/4, 16, PET, and VIC-20

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
1720 PRINT CHR$(147)     :rem 69
2000 OPEN3,4:PRINT#3,T$:PRINT#
     3                  :rem 101
2010 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:IFI/10<>INT(I/10)
     THENPRINT#3," ";:GOTO2030
                        :rem 101
2020 PRINT#3,MID$(STR$(I),2,1)
     ;                  :rem 207
2030 NEXTI:PRINT#3      :rem 106
2040 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:PRINT#3,RIGHT$(ST
     R$(I),1);:NEXTI:PRINT#3
                        :rem 172
2050 FORX=1TOR0:IFX<10THENPRIN
     T#3," ";            :rem 20
2060 PRINT#3,STR$(X)" ";
                         :rem 28
2070 FORY=1TOC0:PRINT#3,MID$(S
     $(X),Y,1);          :rem 98
2080 NEXTY:PRINT#3:NEXTX:PRINT
     #3:PRINT#3:PRINT#3,"WORD
     {SPACE}LIST:"      :rem 201
2090 FORX=1TOW0:IFW$(X)="/"THE
     N2110               :rem 50
2100 PRINT#3,W$(X)      :rem 246
2110 NEXTX:FORI=1TO5:PRINT#3:N
     EXTI:IFSL=0THEN2180
                        :rem 185
2120 PRINT#3,"SOLUTION LIST:":
     PRINT#3,"WORD{21 SPACES}R
     OW{3 SPACES}COLUMN";
                        :rem 213
2130 PRINT#3,"{3 SPACES}DIR"
                        :rem 248
2140 FORX=1TOW0:IFW$(X)="/"THE
     N2170               :rem 52
2150 PRINT#3,W$(X);LEFT$(G$,25
     -LEN(W$(X)));RR(X);LEFT$(
     G$,8-LEN(STR$(RR(X))));
                        :rem 218
2160 PRINT#3,CC(X);LEFT$(G$,6-
     LEN(STR$(CC(X))));FF$(X)
                         :rem 61
2170 NEXTX                :rem 97
2180 CLOSE3:RETURN      :rem 142
```

Program 3: Additional Line Changes For Commodore 16

```
95 MC=60
100 DIM FF$(60),S$(60),W$(60),
    CC(60),RR(60),L(60),E$(2,2
    )
```

Program 4: Additional Line Changes For 8K VIC-20

```
95 MC=50                :rem 160
100 DIM FF$(50),S$(50),W$(50),
    CC(50),RR(50),L(50),E$(2,2
    )                    :rem 25
```

Program 5: Line Changes For Apple

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

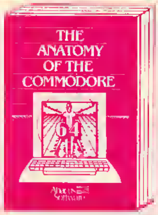
```
38 90 D$ = CHR$ (4):I$ = CHR$ (9
      )
4E 1720 HOME
E1 2000 PRINT D$;"PR#1": PRINT I
        $;"80N"
9E 2010 PRINT T$: PRINT
3F 2020 PRINT "   ";: FOR I = 1
        TO C0: IF I / 10 < > INT
         (I / 10) THEN PRINT " "
        ;: GOTO 2040
1A 2030 PRINT MID$ ( STR$ (I),1,
        1);
77 2040 NEXT I: PRINT
98 2050 PRINT "   ";: FOR I = 1
        TO C0: PRINT RIGHT$ ( ST
        R$ (I),1);: NEXT I: PRIN
        T
C0 2060 FOR X = 1 TO R0: IF X <
        10 THEN PRINT " ";
61 2070 PRINT STR$ (X)"-";
5A 2080 FOR Y = 1 TO C0: PRINT M
        ID$ (S$(X),Y,1);
21 2090 NEXT Y: PRINT : NEXT X:
        PRINT : PRINT : PRINT "W
        ORD LIST:"
3F 2100 FOR X = 1 TO W0: IF W$(X
        ) = "/" THEN 2120
C2 2110 PRINT W$(X)
27 2120 NEXT X: FOR I = 1 TO 5:
        PRINT : NEXT I: IF SL =
        0 THEN 2160
#5 2130 PRINT "SOLUTION LIST:":
        PRINT "WORD
                   ROW   COLUMN   D
        IR": FOR X = 1 TO W0: IF
         W$(X) = "/" THEN 2150
4D 2140 PRINT W$(X) LEFT$ (G$,26
         - LEN (W$(X)))RR(X) LEF
        T$ (G$,9 - LEN ( STR$ (R
        R(X))))CC(X) LEFT$ (G$,6
         - LEN ( STR$ (CC(X))))F
        F$(X)
91 2150 NEXT X
BF 2160 PRINT : PRINT D$;"PR#0":
         RETURN
```

Program 6: IBM PC/PCjr Line Changes

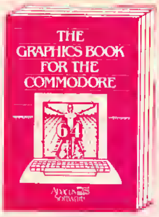
For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
IC 10 DEF SEG=0:POKE 1047,(PEEK(
      1047) OR 64)
JD 20 WIDTH 40:KEY OFF:DEF SEG=&
      H40:RANDOMIZE PEEK(&H6D)
ND 1720 CLS
NF 2000 ON ERROR GOTO 2170
EK 2010 OPEN "LPT1:" FOR OUTPUT
        AS #1:PRINT #1,T$:PRINT
        #1,
MH 2020 PRINT #1,"    ";:FOR I=1
         TO C0:IF I/10<>INT(I/10
        ) THEN PRINT #1," ";:GOT
        O 2040
NH 2030 PRINT #1,MID$(STR$(I),2,
        1);
KE 2040 NEXT I:PRINT #1,
AF 2050 PRINT #1,"    ";:FOR I=1
         TO C0:PRINT #1,RIGHT$(S
        TR$(I),1);:NEXT I:PRINT
        #1,
EH 2060 FOR X=1 TO R0:IF X<10 TH
        EN PRINT #1," ";
PH 2070 PRINT #1,STR$(X)" ";
```

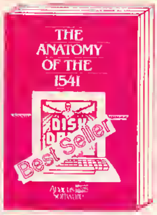
SUPER BOOKS



300 pages, $19.95

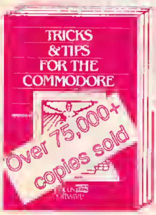


350 pages, $19.95

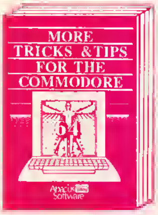


320 pages, $19.95

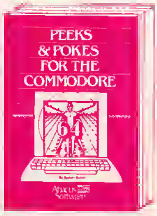
200 pages, $19.95



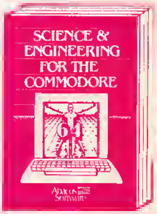
275 pages, $19.95



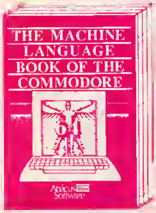
250 pages, $19.95



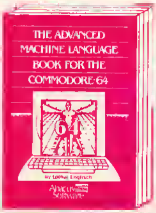
200 pages, $14.95



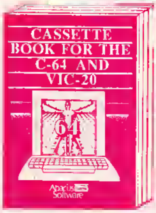
340 pages, $19.95



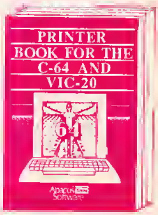
215 pages, $14.95



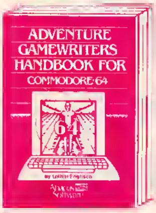
210 pages, $14.95



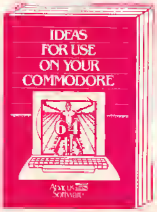
210 pages, $14.95



330 pages, $19.95

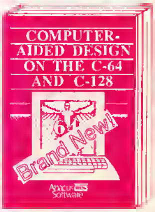


225 pages, $14.95



220 pages, $12.95

250 pages, $19.95



250 pages, $19.95

For fast service call **616/241-5510.** For postage and handling, include $4.00 per order. Foreign orders include $8.00 per item. Money orders and checks in U.S. dollars only. Mastercard, Visa and Amex accepted.

Dealer Inquiries Welcome
More than 1200 dealers nationwide

Abacus Software
P.O. Box 7211 Grand Rapids, MI 49510 616/241-5510

Other software also available!
Call now for free catalog and the name of your nearest dealer. Phone:**616/241-5510.**

Learn All About TELECOMPUTING

COMPUTE!'s
TELECOMPUTING
ON THE
COMMODORE
64

Everything you need to know to start telecomputing on your 64—including special terminal software.

A COMPUTE! Books Publication $12.95

with

COMPUTE!'s Telecomputing on the Commodore 64

This comprehensive guide explains everything you need to know to start telecomputing on your Commodore 64.

Many home computer analysts believe that telecomputing will soon be the most popular use for home computers. *COMPUTE!'s Telecomputing on the Commodore 64* introduces readers to telecommunications, with sections on buying and using modems, accessing information services and bulletin boards, and uploading and downloading files. Best of all, there are terminal programs which allow you to communicate with the information services and bulletin boards.

Edited
$12.95
0-87455-009-2

For your convenience, there's also a disk available which includes all the programs in the book. This 5¼-inch floppy disk saves you hours of typing and prevents many typing errors. It's fast, inexpensive, and ready to load on your Commodore 64. You can order the *Disk* directly from COMPUTE! Books for only $12.95.

To order *COMPUTE!'s Telecomputing on the Commodore 64* or the *Disk*, mail the order form with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403. For fastest service, call toll free 800-334-0868 (In NC 919-275-9809.)

YES! I WANT TO LEARN TO TELECOMPUTE WITH MY COMMODORE 64.

Please send me:

____ ***COMPUTE!'s Telecomputing on the Commodore 64,*** (009-2) $12.95 ea. ________
(Add $2.00 per book U.S. and surface mail, $5.00 airmail, for shipping charges.)

____ ***COMPUTE!'s Telecomputing on the Commodore 64 Disk,*** $12.95 each ________
(Add $2.00 per disk for shipping charges.)

ALL ORDERS
MUST BE
PREPAID IN
U.S. FUNDS

Subtotal ________
NC residents add 4.5% tax ________
Shipping charges ________
Total paid ________

□ Payment enclosed (check or money order)
□ Charge □ MasterCard □ Visa □American Express
Account No. ____________________ Exp. Date ____________
(Required)
Signature ______________________________
Name ______________________________
Address ______________________________
City ____________________ State __________ Zip __________
Please allow 4–6 weeks for delivery 7590091

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

COMPUTE! Books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd.,
1 St. Anne's Road, Eastbourne, East Sussex
BN21 3UN, England.

```
IF 2080 FOR Y=1 TO C0:PRINT #1,MID#(S#(X),Y,1);
CC 2090 NEXT Y:PRINT #1,:NEXT X:PRINT #1,:PRINT #1
        ,:PRINT #1,"WORO LIST:"
DM 2100 FOR X=1 TO W0:IF W#(X)="/" THEN 2120
CI 2110 PRINT #1,W$(X)
HJ 2120 NEXT X:FOR I=1 TO 5:PRINT #1,:NEXT I:IF SL
        =0 THEN 2160
GJ 2130 PRINT #1,"SOLUTION LIST:":PRINT #1,"WORO
                                 ROW   COLUMN   OIR":FOR
         X=1 TO W0:IF W#(X)="/" THEN 2150
LA 2140 PRINT #1,W#(X);LEFT#(G#,25-LEN(W$(X)));RR(
        X);LEFT#(G#,8-LEN(STR#(RR(X))));CC(X);LEFT
        #(G#,6-LEN(STR#(CC(X))));FF#(X)
AG 2150 NEXT X
DA 2160 CLOSE #1:ON ERROR GOTO 0:RETURN
IM 2170 CLOSE #1:PRINT "PRINTER ERROR #";ERR;"OCCU
        RREO.":PRINT "TRY AGAIN."
JL 2180 PRINT:PRINT "HIT A KEY TO CONTINUE"
CA 2190 A#=INKEY$:IF A#="" THEN 2190
HM 2200 RESUME 2010
```

Program 7: TI-99/4A Line Changes

```
80 RANOOMIZE
95 MC=50
100 OIM FF#(50),S#(50),W#(50),CC(50),RR(50),L(5
    0),E$(2,2)
180 G$=G#&" "
510 R1=INT(RND*R0)
520 C1=INT(RNO*C0)
530 O1=INT(RNO*8)+1
570 IF (R1+OX*L(X)<1)+(R1+OX*L(X)>R0)+(C1+OY*L(
    X)<1)THEN 590
640 IF (SEG#(W#(X),Z,1)<"A")+(SEG#(W$(X),Z,1)>"
    Z")THEN 680
670 IF (SEG#(S#(R1),C1,1)<>" ")*(SEG$(S#(R1),C1
    ,1)<>SEG$(W$(X),Z,1))THEN 590
700 IF (SEG#(W#(X),Z,1)<"A")+(SEG#(W$(X),Z,1)>"
    Z")THEN 770
710 S$(R1)=SEG#(S#(R1),1,C1-1)&SEG#(W#(X),Z,1)&
    SEG#(S$(R1),C1+1,LEN(S#(R1))-C1)
840 IF SEG#(X#,1,2)="ST" THEN 1670
850 IF SEG#(X#,1,2)="TR" THEN 500
860 IF SEG$(X#,1,2)<>"SK" THEN 830
910 IF SEG#(S$(X),Y,1)<>" " THEN 930
920 S#(X)=SEG#(S#(X),1,Y-1)&CHR$(INT(26*RNO+65)
    )&SEG#(S#(X),Y+1,LEN(S$(X))-Y)
1190 IF (A#<>"Y")*(A#<>"N")THEN 1180
1550 IF (LEN(X#)<=R0)*(LEN(X#)<=C0)*(X#<>"X")TH
     EN 1610
1690 S#(I)=SEG$(G#,1,C0)
1720 CALL CLEAR
2000 OPEN #1:"RS232"
2010 PRINT #1:T#
2020 PRINT #1
2030 PRINT #1:"{3 SPACES}";
2040 FOR I=1 TO C0
2050 IF I/10=INT(I/10)THEN 2080
2060 PRINT #1:" ";
2070 GOTO 2090
2080 PRINT #1:SEG#(STR#(I),1,1);
2090 NEXT I
2100 PRINT #1
2110 PRINT #1:"{3 SPACES}";
2120 FOR I=1 TO C0
2130 PRINT #1:SEG#(STR$(I),LEN(STR#(I)),1);
2140 NEXT I
2150 PRINT #1
2160 FOR X=1 TO R0
2170 IF X>=10 THEN 2190
2180 PRINT #1:" ";
2190 PRINT #1:STR#(X);" ";
2200 FOR Y=1 TO C0
2210 PRINT #1:SEG$(S$(X),Y,1);
2220 NEXT Y
2230 PRINT #1
2240 NEXT X
2250 PRINT #1
2260 PRINT #1
2270 PRINT #1:"WORO LIST:"
2280 FOR X=1 TO W0
2290 IF W#(X)="/" THEN 2310
2300 PRINT #1:W#(X)
2310 NEXT X
2320 FOR I=1 TO 5
2330 PRINT #1
2340 NEXT I
2350 IF SL=0 THEN 2450
2360 PRINT #1:"SOLUTION LIST:"
2370 PRINT #1:"WORO{21 SPACES}ROW{3 SPACES}COLUM
     N";
2380 PRINT #1:"{3 SPACES}OIR"
2390 FOR X=1 TO W0
2400 IF W$(X)="/" THEN 2440
2410 PRINT #1:W#(X);SEG#(G$,1,25-LEN(W#(X)));RR
     (X);
2420 PRINT #1:SEG#(G#,1,7-LEN(STR#(RR(X))));CC(
     X);SEG#(G#,1,4-LEN(STR#(CC(X))));
2430 PRINT #1:FF#(X)
2440 NEXT X
2450 CLOSE #1
2460 RETURN
```

A SPECIAL APPLE-SPECIFIC PUBLICATION FROM THE PUBLISHERS OF *COMPUTE!*, *COMPUTE!'s Gazette*, AND BEST-SELLING COMPUTE! BOOKS.

Buyer's Guide To Business Software
COMPUTE!'s
Apple
APPLICATIONS SPECIAL
$3.95
Vol. 1 No. 2
Machine Language On The Macintosh: A 68000 Overview
The Office For Everyone: Using Macintosh In Business
Apple At Ten: And What's Coming In The Next Decade
How To Use Bulletin Boards (Or Start Your Own!)
The Electronic Classroom: What To Look For In Education
Ready-To-Use Programs Inside This Issue
Dr. Disk: A Powerful Disk Editor
Worm of Bemer: Fast-paced Action Game
Animation Maker: Create Impressive Graphics
And more

The second semiannual edition of *COMPUTE!'s Apple Applications Special* goes on sale October 1,1985.

Price: $3.95
(in Canada $4.75)

Editorial

COMPUTE!'s Apple Applicatians Special second issue features applicotians, purchosing decisions, tutoriols, ond in-depth feoture articles for awners ond users of Apple persanol computers. There are exciting applicotians far business, schaal, and hame. From softwore ta hardware ta the stote of the industry, this speciol issue serves os o useful tool and a hondy reference. The special issue includes:

Features

Apple at Ten, and What's Coming in the Next Decade: This in-depth laok describes Apple's place in the industry and predicts what it will do in the future. Can the Macintosh Office concept succeed against IBM? How will Apple retain its position in the market when the newest round of computers—such as the Commodore Amiga and Atari ST—reaches homes and schools? This intriguing survey includes comments by computer industry analysts and software manufacturers.

Cruising MAUG: The Micranet Apple Users Graup is probably the best connection any Apple owner can make. Available through CompuServe, MAUG lets Apple users communicate and exchange information and programs. This guide ta MAUG describes just some of its features, and highlights programs from Macintosh desktop utilities to complete terminal software, all of which can be retrieved with a modem.

The Big Picture: Innovative hardware and software can transform the Apple II computer into a powerful graphics machine and enhance the Macintosh's already considerable abilities. Drawing programs, digitizers, and graphics tablets are featured and evaluated in this buyer's guide and tutorial.

Applications

Dr. Disk: Allaws you to read fram, edit, and write ta any block an any disk. An excellent utility which lets you examine disk contents, manipulate catalogs, and even change machine language programs.

Enhanced Applesoft INPUT: A short machine language utility which turns Applesoft INPUT into a more flexible and powerful statement. Allows entry of any valid numeric expression, as well as commas, quotes, or colons as responses to the INPUT prompt.

The Office for Everyone: A major applications feature on using *Word, Chart, File, Multiplan,* and *MacTerminal* on the Macintosh. This tutorial shows how to turn the Macintosh into a powerful business computer.

Apple Electrotype: This simple BASIC program turns any Apple II-series computer into an electronic-style typewriter. Set margins and tabs, underline, and print aut letters, memos, and nates.

Quality editarial in the style and traditian of *COMPUTE!*, the leading magazine of home, educatianal, and recreatianal computing.

There's also o disk ovoiloble which includes oll the progroms from the mogozine. The 5¼-inch floppy disk for the Apple II+, IIe, or IIc costs $16.95 ond is ovoiloble only fram COMPUTE! Publicotians. Send in the ottoched cord with your poyment today ta arder the *Disk.*

Laak far the October issue of *COMPUTE!'s Apple Applications Special* on sole where you buy other COMPUTE! publicotions, or ot Apple computer retoilers. You con olso order directly from COMPUTE! Publicotions.

To order, send in the ottoched cord with your poyment or coll toll-free 800-334-0868 (in NC 919-275-9809.)

Program 8: Atari Version

Version By Patrick Parrish, Programming Supervisor

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
MG 100  NR=99:NW=100:REM NR I
        S MAX # OF RDWS,CDLUM
        NS; NW IS MAX # OF WO
        RDS
MJ 110  OIM G$(NR),FF$(2*NW),
        S$(NR*NR),W$(NW*20),C
        C(NW),RR(NW),L(NW),E$
        (18),O(2,8),A$(5),X$(
        20),T$(30)
CD 120  READ E$:OATA NW NNE W
        {3 SPACES}ESW SSE
KD 130  G$=" ":G$(NR)=G$:G$(2
        )=G$:W$=" ":W$(20*NW)
        =W$:W$(2)=W$
LP 140  FOR I=1 TO 8:READ A,8
        :O(1,I)=A:O(2,I)=8:NE
        XT I:OATA -1,-1,-1,0,
        -1,1,0,-1,0,1,1,-1,1,
        0,1,1
LH 150  X$=" ":X$(20)=X$:X$(2
        )=X$:GOTO 580
DL 160  REM SHELL SORT
BJ 170  PRINT "SORTING...":X=
        1
ML 180  X=2*X:IF X<=W0 THEN 1
        80
MB 190  X=INT(X/2):IF X=0 THE
        N RETURN
BJ 200  FOR Y=1 TO W0-X:Z=Y
MM 210  A=Z+X:IF L(Z)>=L(A) T
        HEN 240
IP 220  X$=W$((Z-1)*20+1,Z*20
        ):W$((Z-1)*20+1,Z*20)
        =W$((A-1)*20+1,A*20):
        W$((A-1)*20+1,A*20)=X
        $
PB 230  8=L(Z):L(Z)=L(A):L(A)
        =8:Z=Z-X:IF Z>0 THEN
        210
DL 240  NEXT Y:GOTO 190
CE 250  REM HIOE WOROS
GP 260  FOR X=1 TD W0
HJ 270  FOR Y=1 TO 50:R1=INT(
        RNO(1)*R0):C1=INT(RNO
        (1)*C0):O1=INT(RNO(1)
        *8)+1:O1=O1
CC 280  OX=O(1,O1):OY=O(2,O1)
        :IF R1+OX*L(X)>=1 AND
         R1+OX*L(X)<=R0 ANO C
        1+DY*L(X)>=1 ANO C1+O
        Y*L(X)<=C0 THEN 310
ID 290  O1=O1*(O1<8)+1:IF O1<
        >O1 THEN 280
DK 300  NEXT Y:GOTO 390
PI 310  FOR Z=1 TO L(X):IF W$
        ((X-1)*20+Z,(X-1)*20+
        Z)<"A" DR W$((X-1)*20
        +Z,(X-1)*20+Z)>"Z" TH
        EN 340
MG 320  R1=R1+DX:C1=C1+DY
MC 330  IF S$((R1-1)*C0+C1,(R
        1-1)*C0+C1)<>" " AND
        S$((R1-1)*C0+C1,(R1-1
        )*C0+C1)<>W$((X-1)*20
        +Z,(X-1)*20+Z) THEN 2
        90
GL 340  NEXT Z:FOR Z=L(X) TO
        1 STEP -1:IF W$((X-1)
        *20+Z,(X-1)*20+Z)<"A"
         OR W$((X-1)*20+Z,(X-
        1)*20+Z)>"Z" THEN 370
MN 350  S$((R1-1)*C0+C1,(R1-1
        )*C0+C1)=W$((X-1)*20+
        Z,(X-1)*20+Z)
NJ 360  RR(X)=R1:CC(X)=C1:FF$
        ((X-1)*2+1,X*2)=E$((D
        X+1)*6+(OY+1)*2+1,(OX
        +1)*6+(OY+1)*2+2):R1=
        R1-DX:C1=C1-DY
DD 370  NEXT Z
DO 380  NEXT X:GDTO 450
EK 390  PRINT "{CLEAR}Sorry,
        but I can't fit word
        number ";STR$(X);" ,
        ";W$((X-1)*20+1,X*20)
        ;" , into the grid."
LJ 400  PRINT "Should I SKip
        it, STart over, or TR
        y again":INPUT X$
GP 410  IF X$(1,2)="ST" THEN
        710
HA 420  IF X$(1,2)="TR" THEN
        270
KB 430  IF X$(1,2)<>"SK" THEN
         400
CM 440  W$((X-1)*20+1,(X-1)*2
        0+1)="/":GOTO 380
PO 450  FOR X=1 TO R0:FOR Y=1
         TO C0:IF S$((X-1)*C0
        +Y,(X-1)*C0+Y)<>" " T
        HEN 470
DH 460  S$((X-1)*C0+Y,(X-1)*C
        0+Y)=CHR$(INT(26*RNO(
        1)+65))
AE 470  NEXT Y:NEXT X
KG 480  REM OONE
HG 490  PRINT :PRINT "I am fi
        nished. What do you w
        ant to call the word
        search":INPUT T$
EJ 500  SL=0:PRINT :PRINT "Do
         you want to print th
        e solution (Y/N)":GOS
        U8 550:IF A$="N" THEN
         520
KD 510  SL=1
CN 520  GOSU8 2000:F=0:PRINT
        "{CLEAR}Oo you want a
        nother grid (Y/N)":GO
        SU8 550:IF A$="N" THE
        N END
IF 530  PRINT :PRINT "Oo you
        want to use the same
        words (Y/N)":GOSU8 55
        0:IF A$="N" THEN 590
FH 540  F=1:GDTO 610
OB 550  INPUT A$:IF A$<>"Y" A
        NO A$<>"N" THEN 550
HL 560  RETURN
KI 570  REM INITIALIZATIDN
LD 580  PRINT CHR$(125):LL=6:
        GOSU8 720:PRINT "
        {12 RIGHT}WORO SEARCH
        ":LL=4:GOSU8 720
FE 590  FOR I=1 TO W0:W$((I-1
        )*20+1,I*20)=G$(1,20)
        :L(I)=0:NEXT I
MP 600  PRINT "How many words
         would you like in yo
        ur word search":INPUT
         W0
IB 610  PRINT :PRINT "How man
        y rows and columns in
         the grid":INPUT R0,C
        0:PRINT
MK 620  IF R0*C0<10*W0 THEN P
        RINT "I don't think I
         could do this.":FDR
        I=1 TD 300:NEXT I:GOT
        O 610
AO 630  PRINT "I think I can
        do this.":IF C0>NR TH
        EN PRINT "(8ut it won
        't fit on the paper.)
        "
KE 640  IF F=1 THEN 710
LJ 650  LL=3:GOSU8 720:PRINT
        "Enter the ";STR$(W0)
        ;" words. To correct
        a mistake, enter X":P
        RINT
GE 660  FOR I=1 TD W0
BM 670  PRINT "Word number ";
        I;":":INPUT X$:IF LEN
        (X$)<=R0 ANO LEN(X$)<
        =C0 ANO X$<>"X" THEN
        700
AD 680  IF X$<>"X" THEN PRINT
         "Oops...the word is
        too long.":GOTO 670
LE 690  I=I-(I>1):GOTO 670
IL 700  L(I)=LEN(X$):W$((I-1)
        *20+1,(I-1)*20+L(I))=
        X$:NEXT I:PRINT CHR$(
        125):GOSU8 170
KF 710  PRINT "{OOWN}Okay, I
        will go to work. Wish
         me luck!":FOR I=1 TO
         R0:S$((I-1)*C0+1,I*C
        0)=G$:NEXT I:GOTO 260
BF 720  FOR I=1 TO LL:PRINT :
        NEXT I:RETURN
AK 1999 REM PRINTER RDUTINE
CI 2000 TRAP 2190:DPEN #1,8,
        0,"P:":PRINT #1;T$:P
        RINT #1
HP 2010 PRINT #1;"
        {3 SPACES}";:FDR I=1
         TO C0:IF I/10<>INT(
        I/10) THEN PRINT #1;
        " ";:GOTO 2030
CA 2020 X$=STR$(I):PRINT #1;
        X$(1,1);
GI 2030 NEXT I:PRINT #1
LI 2040 PRINT #1;"
        {3 SPACES}";:FDR I=1
         TD C0:X$=STR$(I):PR
        INT #1;X$(LEN(X$),LE
        N(X$));:NEXT I:PRINT
         #1
CB 2050 FOR X=1 TO R0:IF X<1
        0 THEN PRINT #1;" ";
GE 2060 PRINT #1;STR$(X);" "
        ;
GC 2070 FOR Y=1 TO C0:PRINT
        #1;S$((X-1)*C0+Y,(X-
        1)*C0+Y);
NA 2080 NEXT Y:PRINT #1:NEXT
         X:PRINT #1:PRINT #1
        :PRINT #1;"WORO LIST
        :"
DE 2090 FOR X=1 TO W0:IF W$(
        (X-1)*20+1,(X-1)*20+
        1)="/" THEN 2110
KK 2100 PRINT #1;W$((X-1)*20
        +1,X*20)
LH 2110 NEXT X:FDR I=1 TO 5:
        PRINT #1:NEXT I:IF S
        L=0 THEN 2180
JD 2120 PRINT #1;"SOLUTION L
        IST:":PRINT #1;"WDRD
        {21 SPACES}ROW
        {3 SPACES}COLUMN
        {3 SPACES}DIR"
OF 2130 FDR X=1 TO W0:IF W$(
        (X-1)*20+1,(X-1)*20+
        1)="/" THEN 2170
PL 2140 PRINT #1;W$((X-1)*20
        +1,X*20);G$(1,6);RR(
        X);
HH 2150 PRINT #1;G$(1,9-LEN(
        STR$(RR(X))));CC(X);
        G$(1,6-LEN(STR$(CC(X
        ))));
IF 2160 PRINT #1;FF$((X-1)*2
        +1,X*2)
GB 2170 NEXT X
BE 2180 CLOSF #1:TRAP 40000:
        RETURN
EB 2190 CLDSE #1:TRAP 40000:
        PRINT "Turn on your
        printer--press RETUR
        N":INPUT X$:GOTO 200
        0
```

©

THE LAST WARRIOR

David Engebretsen

This arcade-style action game was originally written for the IBM PC (with BASICA and color/graphics adapter) and PCjr (with Cartridge BASIC). We've added adaptations for the Commodore 64; Atari 400/800/ XL/XE series (with at least 16K RAM for tape or 24K RAM for disk); and Apple II series. A joystick is required for all versions except the Apple. The Commodore 64 and Atari programs are written completely in machine language.

"Attacked by countless alien ships . . ."

You're the last member of the scouting party sent from Earth. While flying a routine mission, you and your fellow scouts were suddenly attacked by countless alien ships. Your comrades put up a good fight but couldn't survive in the face of the aliens' nonstop shooting. Now the only things between you and utter destruction are your highly advanced force shields and lasers. The aliens may not be as well armed, but they make up for it in sheer numbers. As you blast yet another hostile ship, it is immediately replaced, and your energy supply dwindles....

"The Last Warrior," as you've guessed, is a space shoot-em-up game. The classic object is to destroy as many aliens as possible before they destroy you. Your performance is graded at the end of the game by the number of points you score and by rank: captain, major, colonel, general, or warrior. Scoring and a few other details vary from version to version, but all the programs have one thing in common—the highest ranks are attainable only by the very best players.

IBM Version

After typing the program and saving at least one copy on disk, plug in a joystick and type RUN. Your starfighter appears on the screen, and the program asks you to move the stick to the upper-left corner and press the fire button. Next you're asked to move the stick to the lower-right corner and press the button again. This calibrates the program with your joystick, since different sticks tend to yield different values. (You may also prefer to flip the switches on the bottom of the controller to free the stick from its self-centering mode.)

When the game begins, you find yourself looking out of the front cockpit window at a star field. Below the window is an instrument panel, and an aiming sight floats somewhere on the screen. By maneuvering the sight with the joystick, you can aim your lasers at the alien ships which suddenly appear in view. Press the joystick button to fire shots as the aliens make their passes. With any luck, you'll witness a brilliant explosion as the alien attacker is reduced to stardust. But more aliens soon appear to take his place (up to three at a time), and the battle continues.

Don't fire your lasers indiscriminately, because each shot burns up energy, as indicated by the lower horizontal bar on the instrument panel. This bar shortens toward the left side of the screen as your energy decreases. Alien hits on your force shields also sap energy. The upper horizontal bar on the instrument panel shows the relative number of points you've scored. When this bar goes off the scale toward the right, you advance one rank and the bar starts again at the left. Your rank is constantly displayed on the panel and starts at captain.

The game ends when your ship runs out of energy. Your final rank and score appear on the screen—a higher rank with few points is considered better than a lower rank with many points. Press the joystick button to start another game.

The IBM version of The Last Warrior is written entirely in BASIC and animates the aiming sight and alien ships with the PUT statement. To reduce flickering, one set of variables stores the existing positions of the images while another set holds the new positions. That way, when the program erases an existing image, it can draw the new one immediately without pausing to update the variables. As a result, flickering is hardly noticeable, especially when the program runs on the PC (which is faster than the PCjr).

64 Version

Written entirely in machine language, the 64 version of The Last Warrior must be typed with the "MLX" machine language entry utility found elsewhere in this issue. MLX makes it much easier to enter machine language programs without typos. Be sure you read and understand the instructions for using MLX before entering the data from Program 2.

When you run MLX, you'll be asked for the starting and ending addresses of the program to be entered. For The Last Warrior, the values are:

```
STARTING ADDRESS?    49152
ENDING ADDRESS?      51811
```

If you enter the data from Program 2 in more than one sitting, be sure to use these same values whenever you reload your partially completed work.

After you've finished entering the data and saved at least one copy of the game on disk or tape, load it by typing LOAD"*filename*",8,1 for disk or LOAD"*filename*",1,1 for tape (replace *filename* with whatever name you used for your final version). Next type SYS 49152 and press RETURN. Then plug a joystick into port 2 and push the joystick up to start.

The screen shows the front view from the cockpit with alien ships appearing in the distance against the star field. As the aliens get closer, their ships seem to grow larger. Up to five of them can attack you at once. Move the joystick to aim the floating crosshair and press the button to fire your lasers. Each hit scores 100 points.

The instrument panel at the bottom of the screen shows the level of your ship's shield energy, the number of points you've scored, and a special targeting scope. When the game begins, the energy indicator is set at 5,000 units. Each laser shot you fire depletes the shield energy by 20 units. Alien hits cost 100 units of shield energy. When the energy indicator drops to zero, your shields collapse, leaving you completely vulnerable. The next alien hit will destroy your ship and end the game. At this point, you might as well shoot like crazy, since you're out of shield energy anyway.

To help you hit distant ships, the targeting scope on the instrument panel alerts you when your aiming sight has locked onto an alien. If you press the fire button at this instant, you're guaranteed a direct hit.

When the game ends, the program displays your final score and rank, then waits for you to push the joystick up to start another game. During a game, you can freeze the action by pressing any key, and continue playing by pressing another key.

The 64 version of The Last Warrior uses the multicolor high-resolution graphics screen and all eight sprites for the aiming crosshair, explosion effects, targeting scope image, and maximum of five alien vessels.

Atari Version

Like the 64 version, the Atari adaptation of The Last Warrior is written entirely in machine language and must be typed with the MLX entry utility found elsewhere in this issue. MLX greatly reduces the chances of typos when entering long machine language programs. Be sure you read the instructions and understand how to use Atari MLX before entering data from Program 3.

When you run the MLX program, you'll be asked for starting, ending, and run/init addresses. For The Last Warrior, the proper values are:

STARTING ADDRESS?	8192
ENDING ADDRESS?	10249
RUN/INIT ADDRESS?	8192

If you enter the data from Program 3 in more than one sitting, be sure to use these same values whenever you reload your partially completed work. You'll then be asked whether you wish to create a boot tape, a boot disk, or a disk binary file. For The Last Warrior, you can choose any of these three. However, you should avoid the binary file option if you are not familiar with the procedure for loading`and executing such files.

After you finish entering the data from Program 3, and you've saved at least one copy of The Last Warrior on disk or tape, start the program by loading the boot disk or boot tape or running the binary file created with MLX. For a boot disk, simply insert the disk in the drive and switch on the computer after removing the BASIC cartridge (on a 600XL, 800XL, or XE-series computer, hold down the OPTION button while turning on the machine). To run a boot tape, switch on the computer while holding down the START button (again, remove the BASIC cartridge with a 400, 800, or 1200XL, or simultaneously hold down START and OPTION with a 600XL, 800XL, or XE). Then press the PLAY button on the cassette recorder and hit RETURN. If you used MLX to save the program as a binary disk file, load it with the binary load option in DOS and run at hex address 2000 (decimal 8192).

Plug a joystick into port 1 and press the fire button to start. The screen shows the front view from your ship's cockpit window. Alien vessels first appear as distant dots against the star field, then grow larger as they approach. Their weapons are limited, so they can start shooting at you only at point-blank range. But you can shoot them at any point during their attack. For every alien ship you destroy, you score 100 points; for each hit they make on your energy shield, you lose 100 points of shield energy. You begin the game with 5,000 units of energy, and every shot you fire uses 20 units. (All of this information is indicated on the screen's instrument panel.) You can pause and then continue a game in progress by pressing any key.

All the animation in the Atari version of The Last Warrior is driven by a vertical blank interrupt routine—objects are moved during the split-second interval when the TV's electron beam returns from the lower-right corner of the screen to the upper-left corner to scan another frame. Player/missile graphics are used for the crosshair and alien ships, so no more than three aliens can appear at once. Alien ships actually consist of six separate images which are flipped in succession to create the illusion of an approaching object. The program employs a custom display list to put GRAPHICS 7 at the top of the screen and GRAPHICS 1 at the bottom. The ship's cockpit window is not plotted with the Atari's built-in line-drawing routines, but rather with custom-designed routines which are faster and do not destroy the screen background. Otherwise, laser shots would gradually erase the lines representing the cockpit window.

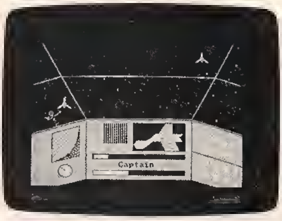
Apple Version

Like the IBM program, the Apple adaptation of The Last Warrior is written in BASIC. However, it does use the HROUT machine language character-plotting routine from "Apple SuperFont" (COMPUTE!, April 1985). All of the alien ships are custom characters created with SuperFont and plotted onto the hi-res graphics screen. The aiming crosshair is drawn with shape tables.

The keyboard controls are programmed in the efficient upside-down T arrangement: I for up, K for down, J for left, and L for right. This is more convenient than the usual I-J-K-M diamond, because you can rest your first three fingers on J-K-L and quickly move your middle finger up and down between I and K.

To fire a laser shot, press the space bar. Press P to pause a game, and press it again to continue.

An instrument panel at the bottom of the cockpit window displays all the important information: points scored (100 for each alien ship you destroy), units of shield energy remaining (the game begins with 5,000), and your current rank. Enemy hits reduce shield energy by 100 units, and your own laser shots cost 20 units each.



An alien ship explodes near the cockpit window while another zooms in for attack in the IBM version of "The Last Warrior."

Program 1: The Last Warrior, IBM Version

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
HB 20 SCREEN 1:COLOR Ø,Ø:CLS:KEY
        OFF:RANDOMIZE TIMER:PLAY"
      mb":STRIG ON
CL 30 DIM SIGHT%(2Ø),SHIP%(5Ø),I
      NFI%(4Ø4),HA%(5Ø),HB%(6Ø),
      HC%(1Ø5),INVER%(1ØØ)
FE 40 REM ** get the images
JJ 5Ø CIRCLE(5,5),3,,,,1:LINE(3,
      3)-(4,4):LINE(7,3)-(6,4):L
      INE(7,7)-(6,6):LINE(3,7)-(
      4,6):GET(2,2)-(8,8),SIGHT%
      :CLS
NJ 6Ø CIRCLE(1Ø,1Ø),1Ø,2:PAINT(1
      Ø,1Ø),2,2:GET(Ø,Ø)-(2Ø,2Ø)
```

With NRI training at home, you can . . .

Move up to a high paying career servicing computers

And you can start by actually building NRI's 16-bit IBM-compatible computer.

You can create your own bright, high paying future as an NRI trained computer service technician. The biggest growth in jobs between now and 1995, according to Department of Labor predictions, will occur in computer service and repair, where demand for trained technicians will double. There is still plenty of room for you to get in on the action—if you get the proper training now.

Total computer systems training, only from NRI

To learn how to work on computers, you have to get inside one. And only NRI takes you inside a computer, with total systems training that gives you hands-on experience with computers, peripherals, and software. As part of your training, you'll build a Sanyo MBC-550-2, which experts have hailed as the "most intriguing" of all the new IBM-compatibles.

Even if you've never had any previous training in electronics, you can succeed with NRI training. You'll start with the basics, rapidly building on the fundamentals of electronics until you master advanced concepts like digital logic, microprocessor design and computer memory. You'll probe into electronic circuits, using the exclusive NRI Discovery Lab® and professional Digital Multimeter, that you keep.

Learn to service today's computers

You'll assemble Sanyo's intelligent keyboard, install the power supply and disk drive, and attach the high resolution monitor—all the while performing hands-on experiments and demonstrations that reinforce your skills.

As you complete your Sanyo, you grasp the "secrets" that qualify you for a new career. You'll learn to program in BASIC and machine language. You'll use utility programs to check out the Sanyo 8088 microprocessor (the same chip used in the IBM PC). And you also get over $1,000 worth of software, including WordStar and CalcStar.

Learn the basics at home

Most importantly, you'll understand the principles common to all computers. Only a person who fully understands all the fundamentals can hope to be able to tackle all computers. NRI makes sure that you'll gain the knowledge and skills to maintain, troubleshoot and service computers.

With NRI training, you'll learn at home on your own time. That means your preparation for a new career or part-time job doesn't have to interfere with your current job. You'll learn at your own pace, in the comfort and convenience of your own home. No classroom pressures, no rigid night school schedules. You're always backed up by the NRI staff and your instructor, who will answer questions, give you guidance and be available for special help if you need it.

Send for free NRI catalog

Let others worry about computers taking their jobs. With NRI training, you'll soon have computers making good paying jobs for you. Send the coupon today for NRI's 100-page catalog, with all the facts about computer training. If the coupon is missing, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

IBM is a Registered Trademark of International Business Machines Corporation.

NRI

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, Washington, DC 20016
We'll give you tomorrow.

McGraw-Hill

For Career courses approved under GI bill,
□ check for details.

□ CHECK ONE FREE CATALOG ONLY

□ Computer Electronics with Microcomputers
□ Data Communications
□ Robotics & Industrial Controls
□ Color TV, Audio, and Video System Servicing
□ Electronic Design Technology
□ Digital Electronics

□ Communications Electronics
□ Industrial Electronics
□ Basic Electronics
□ Telephone Servicing
□ Small Engine Servicing
□ Appliance Servicing

□ Automotive Servicing
□ Air Conditioning, Heating, Refrigeration, & Solar Technology
□ Building Construction
□ Locksmithing & Electronic Security

Name (Please Print) ____________________ Age ______

Street ____________________

City/State/Zip ____________________

Accredited by the National Home Study Council

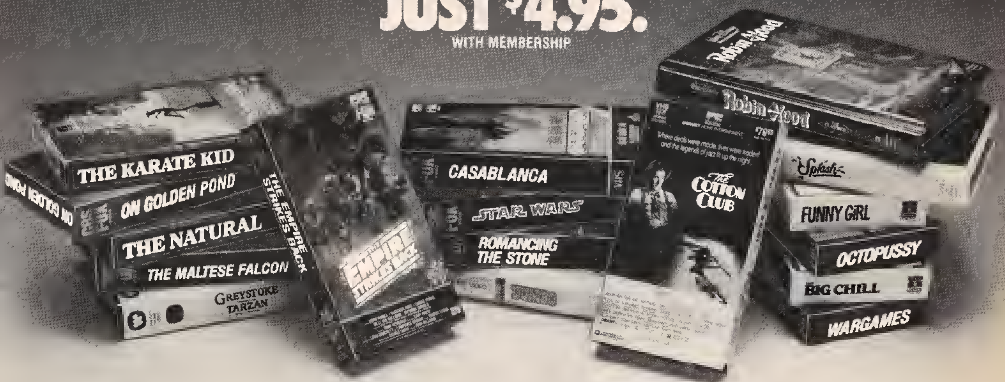
198-095

```
        ,INFI%:CLS
JA 70  LINE(Ø,Ø)-(6Ø,8),3,8F:GET(
       Ø,Ø)-(6Ø,8),INVER%:CLS
ML 8Ø  FOR LOOP=Ø TO 5Ø:READ SHIP
       %(LOOP):NEXT
IO 9Ø  FOR LOOP=Ø TO 5Ø:READ HA%(
       LOOP):NEXT
II 1ØØ FOR LOOP=Ø TO 6Ø:READ H8%
        (LOOP):NEXT
MK 11Ø FOR LOOP=Ø TO 1Ø5:READ HC
        %(LOOP):NEXT
LI 12Ø REM ** set up the screen
OL 13Ø GOSU8 88Ø
BL 14Ø SN=1:SX(1)=16Ø:SY(1)=5Ø:S
        XA(1)=SX(1):SYA(1)=SY(1):
        OLA=1:RANK =Ø:ENE=139:SCO
        =Ø
FB 15Ø GOSU8 137Ø
JP 16Ø SN=1:SX(1)=16Ø:SY(1)=5Ø:S
        XA(1)=SX(1):SYA(1)=SY(1):
        DLA=1
IH 17Ø PUT(127,167),INVER%,PRESE
        T:LOCATE 22,17:PRINT"Capt
        ain":PUT(127,167),INVER%
IC 18Ø XA=Ø:YA=Ø:PUT(XA,YA),SIGH
        T%:PUT(SX(1),SY(1)),SHIP%
MM 19Ø REM ** main program loop
KA 2ØØ GOSU8 29Ø
KP 21Ø GOSU8 56Ø
PP 22Ø IF STRIG(Ø)=-1 THEN GOSU8
         38Ø:V=STRIG(Ø)
ME 23Ø IF RND(1)<.2 THEN PSET(32
        Ø*RND(1),11Ø*RND(1)),3*RN
        D(1)+1
KA 24Ø IF EC>Ø THEN GOSU8 111Ø
CM 25Ø OLA=OLA+.Ø1:OL=INT(OLA)
BO 26Ø GOTO 2ØØ
MJ 27Ø END
CJ 28Ø REM ** JOYSTICK
BF 29Ø X=STICK(Ø):Y=STICK(1):X=X
        -JSX1:Y=Y-JSY1:X=X*TFX:Y=
        Y*TFY
FC 3ØØ IF X<Ø THEN X=Ø
HF 31Ø IF X>313 THEN X=313
HP 32Ø IF Y<Ø THEN Y=Ø
OF 33Ø IF Y>1Ø3 THEN Y=1Ø3
OC 34Ø IF X=Ø AND Y=Ø THEN X=XA:
        Y=YA
KE 35Ø PUT(XA,YA),SIGHT%:PUT(X,Y
        ),SIGHT%:XA=X:YA=Y
MJ 36Ø RETURN
FJ 37Ø REM ** fire !!
FI 38Ø PUT(X,Y),SIGHT%
IM 39Ø FOR P=1 TO SN:PUT(SX(P),S
        Y(P)),8HIP%:NEXT
FM 4ØØ LINE(Ø,11Ø)-(X+3,Y+3),2:L
        INE(319,11Ø)-(X+3,Y+3),2
MJ 41Ø LINE(Ø,11Ø)-(X+3,Y+3),Ø:L
        INE(319,11Ø)-(X+3,Y+3),Ø
FJ 42Ø LINE(Ø,13Ø)-(8Ø,11Ø):LINE
        -(24Ø,11Ø):LINE-(319,13Ø)
GA 43Ø LINE(Ø,6Ø)-(41,5Ø):LINE-(
        28Ø,5Ø):LINE-(319,6Ø)
KA 44Ø LINE(8Ø,11Ø)-(1Ø,Ø):LINE(
        24Ø,11Ø)-(31Ø,Ø)
FO 45Ø IF SX(LOOP)>29Ø THEN SX(L
        OOP)=29Ø
IC 46Ø FOR P=1 TO SN:PUT(SX(P),S
        Y(P)),8HIP%:NEXT
EM 47Ø PUT(X,Y),SIGHT%
JC 48Ø PLAY"l64 t255 bagfedc <ba
        gfedc>"
MP 49Ø SNA=SN
GO 5ØØ FOR LOOP=1 TO SNA
PM 51Ø IF A8S((X+3)-(SX(LOOP)+1Ø
        ))<5 AND A8S((Y+3)-(SY(LO
        OP)+9))<5 THEN EC=EC+1:EX
        (EC)=SX(LOOP):EY(EC)=SY(L
        OOP):OC(EC)=Ø:SN=SN-1:PUT
        (8X(LOOP),SY(LOOP)),SHIP%
        :FOR L=LOOP TO 3:SX(L)=SX
        (L+1):8Y(L)=SY(L+1):SYA(L
        )=8Y(L):SXA(L)=SX(L):NEXT
         L:GOSU8 122Ø
MF 52Ø NEXT
KK 53Ø ENE=ENE-1:IF ENE<=Ø THEN
        GOSUB 15ØØ ELSE LINE(91+E
        NE,18Ø)-(91+ENE,184),Ø
MH 54Ø RETURN
IP 55Ø REM ** enemy ships
IA 56Ø IF RND(1)<.9 THEN GOTO 6Ø
        Ø
BP 57Ø IF SN<3 THEN SN=SN+1:SX(S
        N)=INT(29Ø*RND(1)):SY(SN)
        =INT(1ØØ*RND(1)):PUT(SX(S
        N),SY(SN)),SHIP%:SXA(SN)=
        SX(SN):SYA(SN)=SY(SN):GOT
        O 6ØØ
KM 58Ø IF SN=Ø THEN RETURN
BM 59Ø IF RND(1)>.5 THEN PUT(SX(
        SN),SY(SN)),8HIP%:SN=SN-1
        :IF SN<Ø THEN SN=Ø
LK 6ØØ FOR LOOP=1 TO SN
KG 61Ø GOSU8 29Ø
JP 62Ø IF RND(1)>.95 THEN MX(LOO
        P)=INT(1Ø*RND(1)-5):MY(LO
        OP)=INT(1Ø*RND(1)-5)
MC 63Ø SX(LOOP)=SX(LOOP)+MX(LOOP
        ):SY(LOOP)=SY(LOOP)+MY(LO
        OP)
IK 64Ø IF A8S((X+3)-(SX(LOOP)+1Ø
        ))<3 AND A8S((Y+3)-(SY(LO
        OP)+9))<3 THEN MX(LOOP)=-
        MX(LOOP):IF RND(1)<.5 THE
        N MY(LOOP)=-MY(LOOP)
OL 65Ø IF SX(LOOP)<2 OR SX(LOOP)
        >25Ø THEN MX(LOOP)=-MX(LO
        OP):SX(LOOP)=SX(LOOP)+MX(
        LOOP)
OG 66Ø IF SY(LOOP)<2 OR SY(LOOP)
        >85 THEN MY(LOOP)=-MY(LOO
        P):SY(LOOP)=SY(LOOP)+MY(L
        OOP)
CC 67Ø IF SX(LOOP)<Ø THEN SX(LOO
        P)=Ø
FG 68Ø IF SX(LOOP)>29Ø THEN SX(L
        OOP)=29Ø
FI 69Ø IF SY(LOOP)<Ø THEN SY(LOO
        P)=Ø
PI 7ØØ PUT(SXA(LOOP),SYA(LOOP)),
        SHIP%:PUT(SX(LOOP),SY(LOO
        P)),SHIP%:SXA(LOOP)=SX(LO
        OP):SYA(LOOP)=SY(LOOP)
MF 71Ø NEXT
CA 72Ø IF RND(1)<(OL/2Ø)+SN/1Ø-.
        1 AND SN>Ø THEN GOSU8 75Ø
MH 73Ø RETURN
JO 74Ø REM ** enemy fire
BE 75Ø SN8=INT(SN*RND(1)+1)
BK 76Ø HX=INT(3ØØ*RND(1)):HY=INT
        (85*RND(1)):PUT(X,Y),SIGH
        T%
IH 77Ø FOR P=1 TO SN:PUT(SX(P),S
        Y(P)),SHIP%:NEXT
MF 78Ø PUT(HX,HY),INFI%:LINE(HX+
        1Ø,HY+2)-(SX(SN8)+1Ø,SY(S
        N8)+12),2:LINE-(HX+1Ø,HY+
        18),2
GE 79Ø COLOR 4:PUT(HX,HY),INFI%:
        LINE(HX+1Ø,HY+2)-(SX(SN8)
        +1Ø,8Y(SN8)+12),Ø:LINE-(H
        X+1Ø,HY+18),Ø
OO 8ØØ LINE(Ø,13Ø)-(8Ø,11Ø):LINE
        -(24Ø,11Ø):LINE-(319,13Ø)
        :COLOR Ø
GA 81Ø LINE(Ø,6Ø)-(41,5Ø):LINE-(
        28Ø,5Ø):LINE-(319,6Ø)
KA 82Ø LINE(8Ø,11Ø)-(1Ø,Ø):LINE(
        24Ø,11Ø)-(31Ø,Ø)
OM 83Ø FOR TIM=18Ø TO 2Ø STEP-4:
        SOUND 255-TIM,.1:NEXT
KP 84Ø PUT(X,Y),SIGHT%:FOR P=1 T
        O SN:PUT(SX(P),SY(P)),SHI
        P%:NEXT
HB 85Ø ENE=ENE-4:IF ENE<=Ø THEN
        GOSU8 15ØØ ELSE LINE(91+E
        NE,18Ø)-(229,184),Ø,8F
MO 86Ø RETURN
NP 87Ø REM ** THE SHIP
FL 88Ø FOR LOOP=1 TO 15Ø:PSET(32
        Ø*RND(1),13Ø*RND(1)),3*RN
        D(1)+1:NEXT
GL 89Ø LINE(Ø,13Ø)-(8Ø,11Ø):LINE
        -(24Ø,11Ø):LINE-(319,13Ø)
GP 9ØØ LINE(Ø,6Ø)-(41,5Ø):LINE-(
        28Ø,5Ø):LINE-(319,6Ø)
KP 91Ø LINE(8Ø,11Ø)-(1Ø,Ø):LINE(
        24Ø,11Ø)-(31Ø,Ø)
MA 92Ø LINE(4Ø,199)-(8Ø,19Ø):LIN
        E-(24Ø,19Ø):LINE-(28Ø,199
        )
MC 93Ø LINE(15Ø,116)-(23Ø,153),Ø
        ,8F:LINE(149,115)-(231,15
        4),,8
OE 94Ø PAINT(16Ø,18Ø),3,3
NL 95Ø LINE(Ø,131)-(8Ø,111),Ø:LI
        NE-(24Ø,111),Ø:LINE-(319,
        131),Ø:LINE(8Ø,111)-(8Ø,1
        99),Ø:LINE(24Ø,111)-(24Ø,
        199),Ø
EM 96Ø LINE(9Ø,179)-(23Ø,185),Ø,
        8F:LINE(91,18Ø)-(229,184)
        ,1,8F
OC 97Ø LINE(9Ø,158)-(23Ø,164),Ø,
        8F
OI 98Ø LINE(151,145)-(156,14Ø),1
        :LINE-(17Ø,14Ø),1:LINE-(1
        8Ø,135),1:LINE-(185,131),
        1:LINE-(225,131),1:LINE-(
        22Ø,135),1:LINE-(225,14Ø)
        ,1:LINE-(18Ø,14Ø),1
NN 99Ø LINE-(165,15Ø),1:LINE-(15
        5,15Ø),1:LINE-(151,145),1
        :LINE-(163,145),1:LINE-(1
        68,14Ø),1
GE 1ØØØ LINE (19Ø,131)-(2ØØ,117)
         ,1:LINE-(21Ø,117),1:LINE
         -(21Ø,131),1:LINE(19Ø,13
         5)-(21Ø,135),1:LINE-(22Ø
         ,152),1:LINE-(2ØØ,152),1
         :LINE-(19Ø,135),1:LINE(1
         94,14Ø)-(212,14Ø),Ø
FE 1Ø1Ø PAINT(155,143),3,1:PAINT
         (17Ø,145),CHR$(&H77)+CHR
         $(&HOO),1:PAINT(21Ø,145)
         ,CHR$(&H11)+CHR$(&H44),1
         :PAINT(2Ø5,12Ø),CHR$(&H6
         6)+CHR$(&H99),1
KM 1Ø2Ø FOR LOOP=9Ø TO 14Ø STEP
         15:CIRCLE(LOOP,15Ø),3,1:
         PAINT(LOOP,15Ø),1,1:NEXT
II 1Ø3Ø LINE(1Ø5,143)-(14Ø,117),
         Ø,8F:FOR LOOP=1Ø5 TO 14Ø
          STEP 3:LINE(LOOP,143)-(
         LOOP,117),3:NEXT
DK 1Ø4Ø LO=16Ø:FOR LOOP=7Ø TO 3Ø
          STEP -4:LO=LO+.8:LINE(L
         OOP,LO)-(7Ø,12Ø+(7Ø-LOOP
         )),Ø:NEXT:LINE(3Ø,LO)-(3
         Ø,13Ø),Ø:LINE-(7Ø,12Ø),Ø
         :PAINT(5Ø,14Ø),CHR$(&H66
         )+CHR$(&H99),Ø
JG 1Ø5Ø CIRCLE(5Ø,18Ø),5,1:PAINT
         (5Ø,18Ø),1,1:LINE(5Ø,18Ø
         )-(43,175),Ø:CIRCLE(5Ø,1
         8Ø),1Ø,Ø
EN 1Ø6Ø LO=13Ø:FOR LOOPA=1 TO 2:
         FOR LOOP=26Ø TO 31Ø STEP
          15:LO=LO+4:CIRCLE(LOOP,
         LO),4,1:PAINT(LOOP,LO),1
         ,1:NEXT LOOP:LO=145:NEXT
          LOOPA
KP 1Ø7Ø LINE(24Ø,153)-(319,173),
         Ø
HM 1Ø8Ø LO=16Ø:FOR LOOPA=1 TO 2:
         FOR LOOP=26Ø TO 31Ø STEP
          15:LO=LO+4:LINE(LOOP,LO
         )-(LOOP+6,LO+1),1:LINE-(
         LOOP+6,LO+8),1:LINE-(LOO
         P,LO+7),1:LINE-(LOOP,LO)
         ,1:PAINT(LOOP+2,LO+2),1,
         1:NEXT LOOP:LO=175:NEXT
         LOOPA
JG 1Ø9Ø RETURN
MP 11ØØ REM ** explosion
```

OWN A MOVIE FOREVER FOR THE COST OF SEEING IT ONCE.

JUST $4.95.

WITH MEMBERSHIP



Now you can own any movie here for just $4.95 with membership in the CBS Video Club! That's less than the price of a single movie ticket, less than the price of most rentals, and it's yours for keeps!

Choose from the best, too. *The Empire Strikes Back, Romancing the Stone, Splash* and more. There's no membership fee, and you don't have to buy a lot of movies.

Just two more within the next year. The movies you order will be mailed and billed at regular Club prices which currently range from $39.95 to $79.95 per cassette—plus shipping and handling. (Extra-long films and specials may cost a bit more.)

BONUS PLAN—SAVE 50%

After buying two movies at regular Club prices in the next year, you can cancel. Or stay with us and save even more under our Bonus Plan. With each movie you buy, the plan currently allows you to help yourself to another movie of equal value or less at 50% off. <u>And, right now, save up to $50 more—see the Advance Bonus box above.</u>

About every four weeks (up to 13 times a year) we send you our <u>CBS Video Club Magazine</u>, reviewing our Director's Selection plus many alternate movies.

CHOOSE FROM NUNDREDS OF NIT MOVIES

If you want the Director's Selection, don't do a thing. It will arrive automatically. If you prefer an alternate title, or none at all, just return the card always provided by the date specified.

You'll always have two full weeks to decide. And a toll-free number to call if you have any questions or service requests. (If you ever receive a tape that you had less than two weeks to consider, send it back at our expense.)

Join today and we'll send your movie for just $4.95, along with details of how the Club works. If you're not satisfied, return everything within 10 days for a full, prompt refund —no further obligation.

For faster service, use your credit card and our toll-free number to order. Just call **1-800-457-0866** (in Indiana 1-800-742-1200). Or mail coupon.

60 TOP HITS TO CHOOSE FROM

TITLE	SELECTION NUMBER	TITLE	SELECTION NUMBER	TITLE	SELECTION NUMBER
THE EMPIRE STRIKES BACK	0910092	JANE FONDA'S WORKOUT CHALLENGE	5260042	ROBIN NDOD (Walt Disney)	5299092
ROMANCING THE STDNE	0894092	CASA8LANCA	0507082	EXCALIBUR	6021022
BPLASN	5304022	TOOTSIE	1509042	TIGHTROPE	6051052
THE BIG CHILL	1527022	OUMBO	5251052	BACHELOR PARTY	0926012
WARGAMES	0828002	THE AFRICAN OUEEN	0511022	COUNTRY	5341072
RISKY BUSINESS	6033082	ON GOLDEN POND	0523082	BODY DOUBLE	1713062
THE NATURAL	1649052	THE LONGEST OAY	0577032	REVENGE OF THE NERDS	0925022
STAR WARS	0564162	OIRTY HARRY	6017082	GREYSTOKE—THE LEGENO OF TARZAN, LDRD OF THE APES	6045042
YENTL	0895082	STRIPES	1513082	EDUCATING RITA	1593012
COTTON CLUB	3100032	FUNNY GIRL	1511002	THE ROAD WARRIOR	6028052
CADDYSHACK	6023022	CNRISTINE	1580062	SUPERMAN III	6040092
MAKING MICHAEL JACKSON'S THRILLER	7103012	OCTOPUSSY	0856052	TWILIGNT ZONE—The Movie	6034072
KING KONG (The Original)	5502022	PORKY'8	0775112	ANNIE	1516052
POLICE ACAOEMY	6049002	CLOSE ENCOUNTERS OF THE THIRD KIND—Special Edition	1510012	THE MUPPETS TAKE MANHATTAN	0923042
ARSENIC & OLD LACE	0735102	THE RIGHT STUFF	6043062	ALIEN	0002322
THE COMANCHEROS	0762242	NATIDNAL LAMPDDN'S VACATION	6039022	THE MAGNIFICENT SEVEN	0534212
KARATE KIO	1710092	HIGH ROAD TO CHINA	6022012	THUNDERBALL	0709042
PRIVATE BENJAMIN	6018072	ARTHUR	6024092	BUTCN CASSIDY & THE SUNDANCE KID	0517302
SHE WDRE A YELLOW RI88ON	5504002	RODSTER COGBURN	1018082	PURPLE RAIN	6048012
REO RIVER	7507032	NEVER SAY NEVER AGAIN	6042072	THE MALTESE FALCDN	0508072

Advance Bonus:

SAVE UP TO $50 MORE!

. . . by ordering a second movie right now. Any movie listed in this ad—yours for just $29.95 plus shipping and handling on videocassette. See coupon below.

CBS VIDEO CLUB

1400 North Fruitridge Avenue, Terre Haute, IN 47811

CBS VIDEO CLUB
Dept. 62E, P.O. Box 1111, Terre Haute, IN 47811

Yes, please enroll me in the CBS Video Club under the terms outlined in this advertisement. As a member, I need buy just two more movies at regular Club prices within the next year.

Check one: □ BETA
□ VHS

Send me movie #______________for $4.95

Please check how paying:

□ My check is enclosed. Z54/Z56

□ Charge my introductory movie(s) and future Club purchases to: Z55/Z57

□ MasterCard □ Diners Club
□ American Express □ VISA

Account #____________________

Expiration Date____________________

Signature____________________

□ **Also send me as my Advance Bonus:**

movie #____________ for $29.95 on videocassette plus $3.00 shipping and handling which I'm adding to my above payment.

Name____________________

Address____________________

City______________State______

Zip________Phone (____)__________

Note: CBS Video Club reserves the right to reject any application or cancel any membership. Offer limited to continental U.S. (excluding Alaska). **Applicable sales tax added to all orders.**

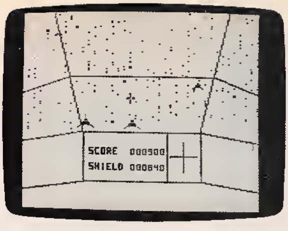
```
NJ 1110 ECA=EC
PB 1120 FOR CO=1 TO EC
QN 1130 IF EX(CO)=Ø AND EY(CO)=Ø
         THEN GOTO 119Ø
AF 1140 IF OC(CO)=Ø THEN PUT(EX(
        CO),EY(CO)),HA%
BN 1150 IF OC(CO)=1 THEN PUT(EX(
        CO),EY(CO)),HA%:PUT(EX(C
        O),EY(CO)),HB%
JN 1160 IF OC(CO)=2 THEN PUT(EX(
        CO),EY(CO)),HB%:PUT(EX(C
        O),EY(CO)),HC%
CB 1170 OC(CO)=OC(CO)+1
DD 1180 IF OC(CO)=4 THEN PUT(EX(
        CO),EY(CO)),HC%:EC=EC-1:
        FOR LO=CO TO EC+1:OC(LO)
        =OC(LO+1):EX(LO)=EX(LO+1
        ):EY(LO)=EY(LO+1):NEXT L
        O:OC(EC+1)=Ø:EX(EC+1)=Ø:
        EY(EC+1) =Ø
BJ 1190 NEXT
IP 1200 RETURN
AE 1210 REM ** scoring for a hit
LI 1220 ENE=ENE+8:IF ENE>139 THE
        N ENE=139
KE 1230 LINE(9Ø+ENE,18Ø)-(83+ENE
        ,184),1,BF:SCO=SCO+3
EC 1240 IF SCO>=136 THEN GOSUB 1
        28Ø
CD 1250 LINE(89+SCO,159)-(91+SCO
        ,163),1,BF
JB 1260 RETURN
CE 1270 REM ** promotion
MA 1280 LINE(9Ø,158)-(23Ø,164),Ø
        ,BF:SCO=3:PUT(127,167),I
        NVER%,PRESET
QP 1290 RANK=RANK+1
AC 1300 IF RANK=1 THEN LOCATE 22
        ,19:PRINT"Major"
JC 1310 IF RANK=2 THEN LOCATE 22
        ,17:PRINT"Colonel"
FL 1320 IF RANK=3 THEN LOCATE 22
        ,17:PRINT"General"
NF 1330 IF RANK=>4 THEN LOCATE 2
        2,17:PRINT"Warrior"
CM 1340 PUT(127,167),INVER%
JA 1350 RETURN
BK 1360 REM ** title page
BD 1370 A=STRIG(Ø)
FA 1380 LOCATE 5,13:PRINT "The L
        ast Warrior"
IE 1390 LOCATE 8,12:PRINT"Move t
        he joystick":LOCATE 9,9:
        PRINT"to the upper-left
        corner ":LOCATE 1Ø,12:PR
        INT"and press button"
CH 1400 IF STRIG(Ø)=-1 THEN JSX1
        =STICK(Ø):JSY1=STICK(1):
        A=STRIG(Ø) ELSE GOTO 14Ø
        Ø
FJ 1410 FOR WAI=1 TO 8ØØ:NEXT WA
        I
NA 1420 LOCATE 9,9:PRINT"to the
        lower-right corner"
IH 1430 IF STRIG(1)=-1 THEN JSX2
        =STICK(Ø):JSY2=STICK(1)
        ELSE GOTO 143Ø
HB 1440 IF JSX2<=JSX1 OR JSY2<JS
        Y1 THEN GOTO 139Ø
GJ 1450 LOCATE 8,12:PRINT SPC(18
        ):LOCATE 9,9:PRINT SPC(2
        5):LOCATE 1Ø,12:PRINT SP
        C(18):LOCATE 5,13:PRINT
        SPC(16):OL=1
GC 1460 TFX=ABS(313/(JSX1-JSX2))
        :TFY=ABS(1Ø3/(JSY1-JSY2)
        )
BF 1470 A=STRIG(Ø)
KL 1480 RETURN
PO 1490 REM ** end
EI 1500 LINE(91,18Ø)-(229,184),Ø
        ,BF
OF 1510 LOCATE 5,16:PRINT"Game O
        ver"
DA 1520 IF RANK=Ø THEN LOCATE 8,
        14:PRINT"Rank: Captain"
LJ 1530 IF RANK=1 THEN LOCATE 8,
        15:PRINT"Rank: Major"
IJ 1540 IF RANK=2 THEN LOCATE 8,
        14:PRINT"Rank: Colonel"
PN 1550 IF RANK=3 THEN LOCATE 8,
        14:PRINT"Rank: General"
PF 1560 IF RANK=>4 THEN LOCATE 8
        ,14:PRINT"Rank: Warrior"
HP 1570 LOCATE 9,16:PRINT"Points
        :";INT(SCO/1.36)
PI 1580 FOR L=1 TO 25
PF 1590 SOUND 25Ø+L*3,.Ø1:SOUND
        215-L*7,.5:SOUND 2ØØ,.1
QM 1600 T=INT(5Ø*RND(1)+2Ø):FOR
        LO=1 TO T:NEXT LO
CC 1610 T=INT(5*RND(1)+4):COLOR
        T
QO 1620 NEXT
FN 1630 COLOR Ø
ND 1640 IF STRIG(1)=Ø THEN 164Ø
DC 1650 FOR LOOP=1 TO SN:PUT(SX(
        LOOP),SY(LOOP)),SHIP%:NE
        XT:PUT(X,Y),SIGHT%
HJ 1660 LINE(91,18Ø)-(229,184),1
        ,BF
LD 1670 LINE(9Ø,158)-(23Ø,164),Ø
        ,BF
EE 1680 RETURN 14Ø
JI 1690 END
NI 1700 DATA 42,15,Ø,2Ø,Ø,Ø,2Ø,Ø
        ,Ø,2Ø,Ø,Ø,2Ø,Ø,Ø,2Ø,Ø,Ø,
        2Ø,Ø,Ø,2Ø,Ø,Ø,65,Ø,256,1
        64Ø4,Ø,256,164Ø4,Ø,5376,
        21569,Ø,215Ø4,5441,Ø,163
        89,276,8Ø,21,2Ø,84,8Ø,Ø,
        5,Ø,Ø,Ø,Ø
KC 1710 DATA 42,15,Ø,4Ø96,Ø,Ø,2Ø
        48Ø,Ø,Ø,16384,Ø,Ø,16384,
        Ø,Ø,16385,Ø,Ø,16389,Ø,Ø,
        424Ø,Ø,Ø,8261,Ø,5376,-28
        582,Ø,215Ø9,2Ø649,Ø,164Ø
        5,5282,Ø,8Ø,1414,Ø,Ø,272
        ,8Ø,Ø,256,8Ø,Ø,Ø,2Ø,Ø,Ø,
        Ø,Ø
KB 1720 DATA 42,19,Ø,Ø,16385,Ø,Ø
        ,5,Ø,Ø,17,Ø,Ø,136,Ø,256,
        16,Ø,256,64,Ø,-23294,Ø,Ø
        ,8454,4Ø,64ØØ,-23984,128
        ,21765,-22174,64,16465,2
        2232,Ø,Ø,1578,64,Ø,1696,
        16,Ø,Ø,8Ø,Ø,Ø,32,Ø,Ø,37,
        Ø,Ø,7,Ø,Ø,5,Ø,Ø,1,Ø,Ø
AB 1730 DATA 6Ø,26,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,-3264Ø,Ø,Ø,Ø,9218
        ,Ø,Ø,Ø,8448,8192,8192,Ø,
        -23552,-32768,Ø,Ø,16386,
        Ø,Ø,Ø,9,Ø,544,Ø,24,Ø,34,
        -325Ø2,96,Ø,256Ø,-229Ø3,
        128,Ø
QJ 1740 DATA -2815Ø,-3Ø552,2,Ø,5
        716,-23932,8,Ø,25Ø89,-21
        872,Ø,Ø,22786,-26112,Ø,Ø
        ,4736,63Ø6,Ø,Ø,512,46Ø8,
        Ø,Ø,512,256Ø,1Ø368,8192,
        512,-32256,2Ø48,Ø,Ø,256Ø
        ,Ø,Ø,Ø,512,128,Ø,128,Ø,1
        48,8,512,Ø,16Ø,32,-32768
        ,Ø,24,Ø,Ø,Ø,32,Ø,Ø,2Ø48,
        Ø,Ø
```

Program 2: The Last Warrior, 64 Version

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article before entering this listing.

```
49152 :162,ØØØ,181,ØØØ,157,Ø99,Ø87
49158 :2Ø2,2Ø2,2Ø8,248,Ø76,137,Ø55
49164 :2Ø1,169,147,Ø32,21Ø,255,ØØ2
```



Enemy ships are approaching your scout vessel in the Commodore 64 version of "The Last Warrior."

```
4917Ø :169,ØØØ,141,17Ø,ØØ2,141,129
49176 :171,ØØ2,141,168,ØØ2,141,137
49182 :169,ØØ2,141,172,ØØ2,133,137
49188 :191,16Ø,Ø23,169,ØØØ,153,22Ø
49194 :ØØØ,212,136,Ø16,248,169,Ø55
492ØØ :Ø47,141,Ø24,212,169,242,115
492Ø6 :141,Ø23,212,169,24Ø,141,212
49212 :Ø13,212,169,128,141,Ø18,229
49218 :212,169,255,141,Ø15,212,Ø46
49224 :169,Ø26,141,ØØ5,212,169,Ø26
4923Ø :ØØ3,141,ØØ1,212,Ø32,Ø82,Ø37
49236 :194,Ø32,Ø65,193,173,Ø98,Ø71
49242 :2Ø2,2Ø8,ØØ9,Ø32,Ø98,194,Ø65
49248 :Ø32,161,194,Ø32,Ø1Ø,196,2Ø9
49254 :Ø32,2Ø6,196,169,ØØ1,141,Ø79
4926Ø :Ø98,2Ø2,Ø32,122,194,169,157
49266 :ØØØ,133,Ø39,Ø32,237,196,239
49272 :Ø32,Ø65,193,169,ØØØ,174,241
49278 :17Ø,ØØ2,172,171,ØØ2,Ø32,163
49284 :192,2ØØ,169,Ø8Ø,166,187,1Ø2
4929Ø :164,188,Ø32,192,2ØØ,Ø32,178
49296 :162,195,Ø32,168,195,Ø32,16Ø
493Ø2 :19Ø,192,Ø32,131,199,Ø32,158
493Ø8 :Ø8Ø,2ØØ,165,197,2Ø1,Ø64,Ø39
49314 :24Ø,215,238,172,ØØ2,165,17Ø
4932Ø :197,2Ø1,Ø64,2Ø8,25Ø,165,229
49326 :197,2Ø1,Ø64,24Ø,25Ø,165,Ø11
49332 :197,2Ø1,Ø64,2Ø8,25Ø,2Ø6,Ø26
49338 :172,ØØ2,24Ø,189,173,ØØØ,194
49344 :22Ø,Ø74,176,Ø1Ø,174,Ø7Ø,148
4935Ø :ØØ3,224,Ø46,24Ø,ØØ3,2Ø6,152
49356 :Ø7Ø,ØØ3,Ø74,176,Ø1Ø,174,199
49362 :Ø7Ø,ØØ3,224,155,24Ø,ØØ3,137
49368 :238,Ø7Ø,ØØ3,Ø74,176,Ø31,Ø4Ø
49374 :174,Ø8Ø,ØØ3,2Ø8,ØØ7,174,1ØØ
4938Ø :Ø6Ø,ØØ3,224,Ø19,24Ø,Ø19,Ø25
49386 :Ø72,173,Ø6Ø,ØØ3,Ø56,233,Ø63
49392 :ØØ1,141,Ø6Ø,ØØ3,173,Ø8Ø,186
49398 :ØØ3,233,ØØØ,141,Ø8Ø,ØØ3,194
494Ø4 :1Ø4,Ø74,176,Ø2Ø,174,Ø8Ø,112
4941Ø :ØØ3,24Ø,ØØ7,174,Ø6Ø,ØØ3,233
49416 :224,Ø7Ø,24Ø,ØØ8,238,Ø6Ø,Ø8Ø
49422 :ØØ3,2Ø8,ØØ3,238,Ø8Ø,ØØ3,Ø37
49428 :Ø74,176,ØØ4,162,ØØ1,134,Ø59
49434 :Ø34,Ø96,169,ØØØ,133,Ø12,214
4944Ø :162,ØØ6,16Ø,Ø12,189,Ø8Ø,129
49446 :ØØ3,Ø74,Ø38,Ø12,189,Ø6Ø,158
49452 :ØØ3,153,ØØØ,2Ø8,189,Ø7Ø,155
49458 :ØØ3,153,ØØ1,2Ø8,136,136,175
49464 :2Ø2,Ø16,233,165,Ø12,141,Ø57
4947Ø :Ø16,2Ø8,Ø96,16Ø,Ø41,185,ØØØ
49476 :Ø45,198,153,ØØØ,ØØ8,185,145
49482 :Ø86,198,153,137,ØØ8,185,Ø73
49488 :128,198,153,198,ØØ8,185,182
49494 :169,198,153,ØØØ,ØØ9,136,239
495ØØ :Ø16,229,169,Ø16,141,Ø89,24Ø
495Ø6 :ØØ8,169,Ø56,141,Ø92,ØØ8,Ø6Ø
49512 :16Ø,Ø5Ø,185,Ø3Ø,2ØØ,153,114
49518 :Ø64,ØØ9,136,Ø16,247,169,239
49524 :Ø32,141,248,ØØ7,169,127,Ø72
4953Ø :141,Ø21,2Ø8,169,1ØØ,141,134
49536 :Ø6Ø,ØØ3,141,Ø7Ø,ØØ3,169,Ø62
49542 :ØØØ,141,Ø8Ø,ØØ3,133,Ø34,Ø13
49548 :162,ØØ7,189,2ØØ,193,157,Ø24
49554 :Ø39,2Ø8,169,ØØØ,157,13Ø,Ø81
4956Ø :ØØ3,2Ø2,Ø16,242,169,ØØØ,Ø16
49566 :141,Ø27,2Ø8,133,Ø13,162,Ø74
```

```
49572 :ØØ5,Ø32,23Ø,197,2Ø2,2Ø8,Ø14
49578 :25Ø,169,Ø37,141,254,ØØ7,ØØ4
49584 :169,136,133,187,169,Ø19,221
4959Ø :133,188,169,185,141,Ø15,245
49596 :2Ø8,169,228,141,Ø14,2Ø8,132
496Ø2 :169,Ø36,141,255,ØØ7,Ø96,13Ø
496Ø8 :Ø1Ø,ØØ3,ØØ4,ØØ5,ØØ6,ØØ7,235
49614 :ØØ1,Ø72,138,Ø72,152,Ø72,2Ø1
4962Ø :169,ØØØ,133,ØØ6,133,ØØ9,15Ø
49626 :152,Ø72,Ø41,ØØ7,133,ØØ4,115
49632 :1Ø4,Ø74,Ø74,Ø74,133,ØØ2,173
49638 :138,Ø72,Ø41,252,Ø1Ø,Ø38,Ø13
49644 :ØØ9,133,ØØ3,1Ø4,Ø41,ØØ3,Ø17
4965Ø :133,ØØ8,169,ØØ3,Ø56,229,Ø72
49656 :ØØ8,168,165,Ø16,192,ØØØ,Ø29
49662 :24Ø,ØØ5,Ø1Ø,Ø1Ø,136,2Ø8,Ø95
49668 :251,133,ØØ8,165,ØØ2,162,213
49674 :ØØ6,Ø1Ø,Ø38,ØØ6,2Ø2,2Ø8,224
4968Ø :25Ø,133,ØØ5,165,ØØ6,Ø24,Ø87
49686 :1Ø1,ØØ2,133,ØØ6,165,ØØ5,178
49692 :1Ø1,ØØ3,133,ØØ5,165,ØØ6,185
49698 :1Ø1,ØØ9,133,ØØ6,165,ØØ5,197
497Ø4 :1Ø1,ØØ4,133,ØØ5,144,ØØ3,174
4971Ø :23Ø,ØØ6,Ø24,1Ø5,ØØØ,133,Ø32
49716 :ØØ5,165,ØØ6,1Ø5,Ø32,133,242
49722 :ØØ6,16Ø,ØØØ,177,ØØ5,166,Ø6Ø
49728 :Ø39,24Ø,ØØ5,ØØ5,ØØ8,Ø76,181
49734 :Ø74,194,Ø69,ØØ8,145,ØØ5,Ø53
4974Ø :1Ø4,168,1Ø4,17Ø,1Ø4,Ø96,Ø54
49746 :169,Ø59,141,Ø17,2Ø8,169,Ø77
49752 :216,141,Ø22,2Ø8,169,Ø29,1Ø5
49758 :141,Ø24,2Ø8,Ø96,169,ØØØ,22Ø
49764 :133,ØØ5,169,ØØ8,133,ØØ6,Ø42
4977Ø :162,Ø56,16Ø,ØØØ,152,145,Ø13
49776 :ØØ5,136,2Ø8,251,23Ø,ØØ6,18Ø
49782 :2Ø2,2Ø8,246,Ø96,169,232,247
49788 :133,ØØ5,169,ØØØ,133,ØØ2,Ø54
49794 :169,ØØ3,133,ØØ6,169,216,Ø58
498ØØ :133,ØØ3,162,ØØ4,16Ø,ØØØ,Ø86
498Ø6 :169,199,145,ØØ5,169,Ø14,Ø75
49812 :145,ØØ2,136,2Ø8,245,23Ø,Ø9Ø
49818 :ØØ6,23Ø,ØØ3,2Ø2,2Ø8,238,Ø17
49824 :Ø96,162,Ø18,16Ø,ØØØ,169,253
4983Ø :ØØØ,133,ØØ2,169,Ø32,133,123
49836 :ØØ3,173,Ø27,212,2Ø1,Ø2Ø,Ø4Ø
49Ø42 :176,ØØ7,173,Ø27,212,Ø41,Ø46
49848 :ØØ3,2Ø8,ØØ2,169,ØØØ,145,199
49854 :ØØ2,136,2Ø8,235,23Ø,ØØ3,236
4986Ø :2Ø2,2Ø8,23Ø,Ø96,133,Ø14,Ø55
49866 :Ø72,138,Ø72,152,Ø72,134,Ø74
49872 :Ø1Ø,134,Ø18,132,Ø11,132,133
49878 :Ø2Ø,169,ØØØ,133,Ø17,133,174
49884 :Ø19,165,Ø21,133,Ø15,165,226
4989Ø :Ø14,197,Ø1Ø,176,Ø16,169,Ø4Ø
49896 :ØØ1,141,Ø9Ø,ØØ3,165,Ø1Ø,13Ø
499Ø2 :Ø56,229,Ø14,141,11Ø,ØØ3,Ø23
499Ø8 :Ø76,ØØ4,195,169,ØØØ,141,Ø61
49914 :Ø9Ø,ØØ3,165,Ø14,Ø56,229,Ø39
4992Ø :Ø1Ø,141,11Ø,ØØ3,165,Ø15,188
49926 :197,Ø11,176,Ø42,169,ØØ1,Ø9Ø
49932 :141,1ØØ,ØØ3,165,Ø11,Ø56,232
49938 :229,Ø15,141,12Ø,ØØ3,Ø76,Ø9Ø
49944 :Ø52,195,169,ØØØ,141,1ØØ,169
4995Ø :ØØ3,165,Ø15,Ø56,229,Ø11,253
49956 :141,12Ø,ØØ3,173,11Ø,ØØ3,Ø74
49962 :2Ø5,12Ø,ØØ3,176,ØØ3,173,21Ø
49968 :12Ø,ØØ3,133,Ø37,165,Ø18,Ø12
49974 :17Ø,165,Ø2Ø,164,Ø39,2Ø8,Ø52
4998Ø :Ø12,197,Ø38,2Ø8,ØØ4,228,235
49986 :193,24Ø,ØØ8,133,Ø38,134,Ø44
49992 :193,168,Ø32,2Ø7,193,173,Ø14
49998 :Ø9Ø,ØØ3,2Ø8,Ø17,165,Ø17,Ø66
5ØØØ4 :Ø24,1Ø9,11Ø,ØØ3,133,Ø17,224
5ØØ1Ø :165,Ø18,1Ø5,ØØØ,133,Ø18,Ø17
5ØØ16 :Ø76,113,195,165,Ø17,Ø56,2Ø6
5ØØ22 :237,11Ø,ØØ3,133,Ø17,165,255
5ØØ28 :Ø18,233,ØØØ,133,Ø18,173,171
5ØØ34 :1ØØ,ØØ3,2Ø8,Ø17,165,Ø19,114
5ØØ4Ø :Ø24,1Ø9,12Ø,ØØ3,133,Ø19,Ø16
5ØØ46 :165,Ø2Ø,1Ø5,ØØØ,133,Ø2Ø,Ø57
5ØØ52 :Ø76,149,195,165,Ø19,Ø56,Ø24
5ØØ58 :237,12Ø,ØØ3,133,Ø19,165,Ø47
5ØØ64 :Ø2Ø,233,ØØØ,133,Ø2Ø,238,Ø2Ø
5ØØ7Ø :Ø22,212,198,Ø37,2Ø8,152,211
5ØØ76 :1Ø4,168,1Ø4,17Ø,1Ø4,Ø96,134
5ØØ82 :16Ø,ØØØ,136,2Ø8,253,Ø96,247
5ØØ88 :165,Ø34,24Ø,Ø93,169,ØØ2,1Ø3
5ØØ94 :133,Ø16,169,ØØ6,141,ØØ8,135
5Ø1ØØ :212,169,Ø33,141,Ø11,212,19Ø
```

```
5Ø1Ø6 :173,Ø6Ø,ØØ3,Ø56,233,Ø12,211
5Ø112 :133,ØØ2,173,Ø8Ø,ØØ3,233,Ø48
5Ø11Ø :ØØØ,133,ØØ3,Ø7Ø,ØØ3,1Ø2,253
5Ø124 :ØØ2,162,ØØØ,16Ø,121,173,Ø54
5Ø13Ø :Ø7Ø,ØØ3,Ø56,233,Ø4Ø,133,233
5Ø136 :Ø21,165,ØØ2,Ø32,2ØØ,194,Ø62
5Ø142 :162,159,Ø32,2ØØ,194,162,1Ø7
5Ø148 :ØØØ,Ø32,2ØØ,194,162,159,2Ø7
5Ø154 :Ø32,2ØØ,194,169,ØØØ,133,194
5Ø16Ø :Ø34,141,ØØ8,212,165,187,219
5Ø166 :Ø56,233,Ø2Ø,133,187,165,Ø16
5Ø172 :188,233,ØØØ,133,188,Ø16,242
5Ø178 :ØØ6,169,ØØØ,133,187,133,118
5Ø184 :188,Ø96,169,ØØØ,141,Ø33,123
5Ø19Ø :2Ø8,169,ØØ1,133,Ø16,133,162
5Ø196 :Ø39,162,Ø4Ø,16Ø,121,169,199
5Ø2Ø2 :12Ø,133,Ø21,169,12Ø,Ø32,1Ø9
5Ø2Ø8 :2ØØ,194,162,Ø4Ø,16Ø,121,141
5Ø214 :169,12Ø,133,Ø21,169,12Ø,ØØ2
5Ø22Ø :Ø32,2ØØ,194,162,Ø4Ø,16Ø,Ø64
5Ø226 :12Ø,169,ØØØ,133,Ø21,169,15Ø
5Ø232 :Ø2Ø,Ø32,2ØØ,194,162,12Ø,Ø16
5Ø238 :16Ø,12Ø,169,ØØØ,133,Ø21,153
5Ø244 :169,14Ø,Ø32,2ØØ,194,162,197
5Ø25Ø :Ø3Ø,16Ø,Ø65,169,Ø64,133,183
5Ø256 :Ø21,169,13Ø,Ø32,2ØØ,194,Ø58
5Ø262 :162,ØØØ,16Ø,Ø8Ø,169,Ø65,21Ø
5Ø268 :133,Ø21,169,Ø3Ø,Ø32,2ØØ,165
5Ø274 :194,162,159,16Ø,ØØØ,169,254
5Ø28Ø :Ø64,133,Ø21,169,13Ø,Ø32,141
5Ø286 :2ØØ,194,162,ØØØ,16Ø,13Ø,18Ø
5Ø292 :169,12Ø,133,Ø21,169,Ø4Ø,ØØØ
5Ø298 :Ø32,2ØØ,194,162,159,169,Ø14
5Ø3Ø4 :12Ø,Ø32,2ØØ,194,162,12Ø,188
5Ø31Ø :16Ø,17Ø,169,12Ø,133,Ø21,139
5Ø316 :Ø32,2ØØ,194,162,159,16Ø,Ø23
5Ø322 :1ØØ,169,17Ø,133,Ø21,169,22Ø
5Ø328 :12Ø,Ø32,2ØØ,194,162,Ø4Ø,132
5Ø334 :16Ø,171,Ø32,2ØØ,194,169,Ø6Ø
5Ø34Ø :12Ø,133,Ø21,169,Ø4Ø,Ø32,167
5Ø346 :2ØØ,194,132,Ø21,13Ø,162,249
5Ø352 :ØØØ,16Ø,18Ø,Ø32,2ØØ,194,174
5Ø358 :162,Ø97,16Ø,17Ø,169,12Ø,Ø36
5Ø364 :133,Ø21,138,Ø32,2ØØ,194,138
5Ø37Ø :162,1ØØ,16Ø,16Ø,169,13Ø,Ø59
5Ø376 :133,Ø21,13Ø,Ø32,2ØØ,194,15Ø
5Ø382 :162,Ø99,16Ø,145,169,144,Ø61
5Ø388 :133,Ø21,169,118,Ø32,2ØØ,117
5Ø394 :194,16Ø,Ø47,185,Ø35,199,Ø14
5Ø4ØØ :153,152,Ø53,185,ØØ3,199,Ø25
5Ø4Ø6 :153,Ø24,Ø56,136,Ø16,241,Ø88
5Ø412 :Ø96,12Ø,169,25Ø,141,Ø2Ø,ØØ8
5Ø418 :ØØ3,169,196,141,Ø21,ØØ3,ØØ7
5Ø424 :Ø88,Ø96,173,172,ØØ2,24Ø,251
5Ø43Ø :ØØ3,Ø76,Ø49,234,162,ØØ5,Ø15
5Ø436 :189,Ø9Ø,ØØ3,2Ø8,Ø42,189,213
5Ø442 :14Ø,ØØ3,Ø24,125,11Ø,ØØ3,159
5Ø448 :157,14Ø,ØØ3,1Ø9,Ø6Ø,ØØ3,Ø56
5Ø454 :1Ø5,ØØØ,157,Ø6Ø,ØØ3,189,Ø24
5Ø46Ø :Ø8Ø,ØØ3,ØØ8,1Ø5,ØØØ,157,125
5Ø466 :Ø8Ø,ØØ3,Ø4Ø,144,Ø52,1Ø9,Ø3Ø
5Ø472 :Ø6Ø,ØØ3,2Ø1,Ø6Ø,144,Ø45,Ø41
5Ø478 :Ø32,23Ø,197,144,Ø4Ø,189,11Ø
5Ø484 :14Ø,ØØ3,Ø56,253,11Ø,ØØ3,1Ø5
5Ø49Ø :157,14Ø,ØØ3,1Ø9,Ø6Ø,ØØ3,Ø98
5Ø496 :233,ØØØ,157,Ø6Ø,ØØ3,189,194
5Ø5Ø2 :Ø8Ø,ØØ3,ØØ8,233,ØØØ,157,Ø39
5Ø5Ø8 :Ø8Ø,ØØ3,Ø4Ø,176,Ø1Ø,189,Ø62
5Ø514 :Ø6Ø,ØØ3,2Ø1,Ø3Ø,176,ØØ3,Ø43
5Ø52Ø :Ø32,23Ø,197,189,1ØØ,ØØ3,Ø71
5Ø526 :2Ø8,Ø31,189,15Ø,ØØ3,Ø24,187
5Ø532 :125,12Ø,ØØ3,157,15Ø,ØØ3,146
5Ø538 :189,Ø7Ø,ØØ3,1Ø5,ØØØ,157,118
5Ø544 :Ø7Ø,ØØ3,2Ø1,157,176,ØØ4,211
5Ø55Ø :2Ø1,Ø2Ø,176,Ø31,Ø32,23Ø,Ø4Ø
5Ø556 :197,144,Ø26,189,15Ø,ØØ3,Ø65
5Ø562 :Ø56,253,12Ø,ØØ3,157,15Ø,1Ø1
5Ø568 :ØØ3,189,Ø7Ø,ØØ3,233,ØØØ,122
5Ø574 :157,Ø7Ø,ØØ3,2Ø1,Ø2Ø,144,225
5Ø58Ø :229,2Ø1,157,176,225,222,Ø78
5Ø586 :13Ø,ØØ3,2Ø8,Ø6Ø,169,Ø4Ø,252
5Ø592 :157,13Ø,ØØ3,189,248,ØØ7,126
5Ø598 :2Ø1,Ø36,2Ø8,Ø45,165,191,244
5Ø6Ø4 :2Ø8,Ø44,173,Ø27,212,2Ø1,Ø13
5Ø61Ø :Ø9Ø,176,Ø37,169,ØØ1,133,Ø16
5Ø616 :191,189,Ø6Ø,ØØ3,Ø56,233,148
5Ø622 :Ø12,133,253,189,Ø8Ø,ØØ3,Ø92
5Ø628 :233,ØØØ,133,254,Ø7Ø,254,116
5Ø634 :1Ø2,253,189,Ø7Ø,ØØ3,Ø56,1Ø7
```

```
5Ø64Ø :233,Ø4Ø,133,254,Ø76,218,138
5Ø646 :197,254,248,ØØ7,2Ø2,24Ø,Ø82
5Ø652 :ØØ3,Ø76,ØØ4,197,Ø32,Ø28,Ø48
5Ø658 :193,Ø76,Ø49,234,173,Ø27,21Ø
5Ø664 :212,157,Ø6Ø,ØØ3,173,Ø27,Ø96
5Ø67Ø :212,Ø74,157,Ø7Ø,ØØ3,16Ø,146
5Ø676 :ØØØ,173,Ø27,212,Ø16,ØØ2,162
5Ø682 :16Ø,ØØ1,152,157,Ø9Ø,ØØ3,Ø45
5Ø688 :16Ø,ØØØ,173,Ø27,212,Ø16,Ø76
5Ø694 :ØØ2,16Ø,ØØ1,152,157,1ØØ,Ø66
5Ø7ØØ :ØØ3,173,Ø27,212,Ø74,Ø24,Ø13
5Ø7Ø6 :1Ø5,Ø5Ø,157,12Ø,ØØ3,173,114
5Ø712 :Ø27,212,Ø74,Ø24,1Ø5,Ø5Ø,ØØ4
5Ø718 :157,11Ø,ØØ3,169,1ØØ,157,214
5Ø724 :13Ø,ØØ3,169,Ø33,157,248,ØØ8
5Ø73Ø :ØØ7,Ø24,Ø96,ØØØ,ØØØ,ØØØ,169
5Ø736 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,Ø48
5Ø742 :ØØØ,ØØØ,ØØØ,ØØØ,Ø24,ØØØ,Ø78
5Ø748 :ØØØ,Ø24,ØØØ,ØØØ,Ø24,ØØØ,1Ø8
5Ø754 :ØØØ,Ø24,ØØØ,ØØ3,195,192,224
5Ø76Ø :ØØ3,195,192,ØØØ,Ø24,ØØØ,23Ø
5Ø766 :ØØØ,Ø24,ØØØ,ØØØ,Ø24,ØØØ,126
5Ø772 :ØØØ,Ø24,ØØØ,ØØØ,ØØØ,ØØØ,1Ø8
5Ø778 :ØØØ,ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,Ø98
5Ø784 :ØØ8,ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,112
5Ø79Ø :Ø2Ø,ØØØ,ØØØ,Ø62,ØØØ,ØØØ,184
5Ø796 :213,128,ØØ1,ØØØ,Ø64,ØØØ,ØØ2
5Ø8Ø2 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,114
5Ø8Ø8 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,12Ø
5Ø814 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,126
5Ø82Ø :ØØ8,ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,148
5Ø826 :ØØ8,ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,154
5Ø832 :Ø28,ØØØ,ØØØ,Ø34,ØØØ,ØØØ,2Ø6
5Ø838 :127,ØØØ,ØØØ,127,ØØØ,ØØ1,149
5Ø844 :162,192,ØØ6,ØØØ,Ø48,ØØØ,Ø52
5Ø85Ø :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,162
5Ø856 :ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,ØØ8,184
5Ø862 :ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,ØØ8,19Ø
5Ø868 :ØØØ,ØØØ,ØØ8,ØØØ,ØØØ,Ø28,216
5Ø874 :ØØØ,ØØØ,Ø62,ØØØ,ØØØ,Ø65,Ø57
5Ø88Ø :ØØØ,ØØØ,255,128,ØØ1,255,Ø63
5Ø886 :192,ØØ1,255,192,ØØ6,193,Ø13
5Ø892 :176,Ø24,ØØØ,Ø12,Ø96,ØØØ,ØØØ
5Ø898 :ØØ3,ØØØ,Ø84,Ø68,Ø68,Ø68,245
5Ø9Ø4 :Ø68,Ø84,ØØØ,ØØØ,Ø16,Ø16,144
5Ø91Ø :Ø16,Ø16,Ø16,Ø16,ØØØ,ØØØ,Ø3Ø
5Ø916 :Ø84,ØØ4,Ø84,Ø64,Ø64,Ø84,1ØØ
5Ø922 :ØØØ,ØØØ,Ø84,ØØ4,Ø84,ØØ4,154
5Ø928 :ØØ4,Ø84,ØØØ,ØØØ,Ø68,Ø68,2Ø8
5Ø934 :Ø84,ØØ4,ØØ4,ØØ4,ØØØ,ØØØ,Ø86
5Ø94Ø :Ø84,Ø64,Ø84,ØØ4,ØØ4,Ø84,Ø64
5Ø946 :ØØØ,ØØØ,Ø84,Ø64,Ø84,Ø68,Ø46
5Ø952 :Ø68,Ø84,ØØØ,ØØØ,Ø84,ØØ4,248
5Ø958 :ØØ4,ØØ4,ØØ4,ØØ4,ØØØ,ØØØ,Ø3Ø
5Ø964 :Ø84,Ø68,Ø84,Ø68,Ø68,Ø84,22Ø
5Ø97Ø :ØØØ,ØØØ,Ø84,Ø68,Ø84,ØØ4,Ø1Ø
5Ø976 :ØØ4,Ø84,ØØØ,168,128,128,Ø32
5Ø982 :168,ØØ8,ØØ8,168,ØØØ,168,Ø46
5Ø988 :128,128,128,128,128,168,Ø84
5Ø994 :ØØØ,168,136,136,136,136,25Ø
51ØØØ :136,168,ØØØ,168,136,136,Ø32
51ØØ6 :168,16Ø,136,136,ØØØ,168,Ø62
51Ø12 :128,128,16Ø,128,128,168,14Ø
51Ø18 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,Ø74
51Ø24 :ØØØ,ØØØ,ØØØ,252,192,192,2Ø4
51Ø3Ø :252,Ø12,Ø12,252,ØØØ,2Ø4,Ø5Ø
51Ø36 :2Ø4,2Ø4,252,2Ø4,2Ø4,2Ø4,Ø84
51Ø42 :ØØØ,Ø48,Ø48,Ø48,Ø48,Ø48,Ø82
51Ø48 :Ø48,Ø48,ØØØ,252,192,192,Ø68
51Ø54 :24Ø,192,192,252,ØØØ,192,154
51Ø6Ø :192,192,192,192,192,252,Ø48
51Ø66 :ØØØ,24Ø,2Ø4,2Ø4,2Ø4,2Ø4,154
51Ø72 :2Ø4,24Ø,ØØØ,198,Ø13,2Ø8,223
51Ø78 :ØØ8,169,ØØØ,141,Ø66,ØØ3,ØØ9
51Ø84 :141,Ø86,ØØ3,162,ØØ5,169,194
51Ø9Ø :224,Ø56,125,248,ØØ7,133,171
51Ø96 :ØØ2,Ø73,255,133,ØØ3,173,Ø23
511Ø2 :Ø6Ø,ØØ3,Ø56,253,Ø6Ø,ØØ3,Ø81
511Ø8 :197,ØØ3,176,ØØ4,197,ØØ2,231
51114 :176,Ø17,173,Ø7Ø,ØØ3,Ø56,153
5112Ø :253,Ø7Ø,ØØ3,197,ØØ3,176,11Ø
51126 :Ø15,197,ØØ2,176,ØØ2,144,2Ø6
51132 :ØØ9,2Ø2,2Ø8,221,169,127,1ØØ
51138 :141,Ø21,2Ø8,Ø96,169,255,Ø6Ø
51144 :141,Ø21,2Ø8,165,Ø34,24Ø,241
5115Ø :246,189,Ø6Ø,ØØ3,141,Ø66,143
51156 :ØØ3,189,Ø7Ø,ØØ3,141,Ø76,182
51162 :ØØ3,169,Ø2Ø,133,Ø13,169,213
51168 :128,141,ØØ4,212,169,129,239
51174 :141,ØØ4,212,189,Ø8Ø,ØØ3,Ø91
```

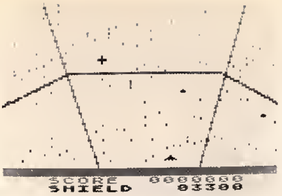
```
51180 :141,Ø86,ØØ3,173,17Ø,ØØ2,Ø43
51186 :Ø24,1Ø5,1ØØ,141,17Ø,ØØ2,Ø16
51192 :173,171,ØØ2,1Ø5,ØØØ,141,Ø72
51198 :171,ØØ2,173,169,ØØ2,2Ø1,2Ø4
512Ø4 :ØØ4,24Ø,Ø18,238,168,ØØ2,162
5121Ø :173,168,ØØ2,2Ø1,Ø19,2Ø8,Ø13
51216 :ØØ8,169,ØØØ,141,168,ØØ2,248
51222 :238,169,ØØ2,Ø32,23Ø,197,122
51228 :144,167,ØØØ,146,ØØØ,ØØ2,231
51234 :ØØ4,ØØØ,ØØ9,Ø33,Ø32,ØØØ,112
5124Ø :142,ØØØ,Ø18,Ø56,Ø72,ØØ5,Ø77
51246 :Ø65,ØØØ,Ø65,Ø1Ø,212,Ø22,164
51252 :121,192,ØØ2,Ø42,Ø68,Ø84,Ø49
51258 :148,168,ØØ1,162,Ø66,Ø2Ø,111
51264 :2Ø8,144,ØØ6,1ØØ,ØØ4,ØØ1,Ø15
5127Ø :136,192,ØØ2,Ø36,ØØ8,ØØØ,188
51276 :128,128,ØØØ,Ø4Ø,165,191,216
51282 :24Ø,1Ø7,166,253,224,159,2Ø7
51288 :176,Ø92,164,254,192,Ø4Ø,238
51294 :144,Ø86,192,15Ø,176,Ø82,156
513ØØ :169,ØØ3,133,Ø16,152,Ø56,117
513Ø6 :233,Ø3Ø,133,Ø21,165,253,173
51312 :1Ø9,Ø27,212,2Ø1,14Ø,176,2Ø9
51318 :247,2Ø1,Ø2Ø,144,243,169,118
51324 :129,141,Ø11,212,141,ØØ8,254
5133Ø :212,Ø32,2ØØ,194,Ø72,169,241
51336 :Ø1Ø,141,Ø33,2Ø8,1Ø4,Ø32,152
51342 :2ØØ,194,Ø32,162,195,169,Ø7Ø
51348 :ØØØ,141,Ø33,2Ø8,165,187,114
51354 :ØØ5,188,2Ø8,ØØ3,Ø76,Ø63,185
5136Ø :2Ø1,165,187,Ø56,233,1ØØ,Ø78
51366 :133,187,165,188,233,ØØØ,Ø48
51372 :133,188,Ø16,ØØ6,169,ØØØ,172
51378 :133,187,133,188,169,ØØØ,22Ø
51384 :133,191,169,Ø16,141,Ø11,Ø77
5139Ø :212,Ø96,133,ØØ2,138,Ø72,Ø75
51396 :169,ØØØ,133,ØØ3,162,ØØ3,154
514Ø2 :ØØ6,ØØ2,Ø38,ØØ3,2Ø2,2Ø8,149
514Ø8 :249,165,ØØ2,Ø24,1Ø5,2Ø8,193
51414 :141,Ø46,2Ø1,165,ØØ3,1Ø5,1Ø7
5142Ø :Ø53,141,Ø47,2Ø1,1Ø4,133,131
51426 :247,132,248,169,ØØØ,133,131
51432 :249,133,25Ø,133,251,162,13Ø
51438 :Ø15,ØØ6,247,Ø38,248,12Ø,144
51444 :248,165,249,1Ø1,249,133,1Ø9
5145Ø :249,165,25Ø,1Ø1,25Ø,133,118
51456 :25Ø,165,251,1Ø1,251,133,127
51462 :251,216,Ø88,2Ø2,Ø16,227,238
51468 :162,ØØ2,181,249,Ø72,Ø74,24Ø
51474 :Ø74,Ø74,Ø74,Ø32,Ø34,2Ø1,251
5148Ø :1Ø4,Ø41,Ø15,Ø32,Ø34,2Ø1,195
51486 :2Ø2,Ø16,237,Ø96,134,ØØ2,2Ø5
51492 :Ø1Ø,Ø1Ø,Ø1Ø,168,162,ØØ7,147
51498 :185,211,198,141,255,255,ØØ7
515Ø4 :238,Ø46,2Ø1,2Ø8,ØØ3,238,214
5151Ø :Ø47,2Ø1,2ØØ,2Ø2,Ø16,238,19Ø
51516 :166,ØØ2,Ø96,162,255,154,127
51522 :169,128,141,ØØ4,212,141,Ø93
51528 :Ø11,212,169,129,141,ØØ4,226
51534 :212,162,ØØØ,142,Ø66,ØØ3,151
5154Ø :142,Ø76,ØØ3,189,Ø99,2Ø2,Ø27
51546 :149,ØØØ,2Ø2,2Ø8,248,162,Ø35
51552 :15Ø,173,Ø27,212,Ø41,ØØ7,194
51558 :133,ØØ2,173,Ø17,2Ø8,Ø41,164
51564 :248,ØØ5,ØØ2,141,Ø17,2Ø8,217
5157Ø :141,Ø22,2Ø8,141,Ø33,2Ø8,Ø99
51576 :165,162,197,162,24Ø,252,Ø18
51582 :142,ØØ1,212,2Ø2,2Ø8,221,Ø88
51588 :169,Ø16,141,ØØ4,212,Ø32,194
51594 :129,255,Ø32,138,255,169,Ø92
516ØØ :Ø27,141,Ø17,2Ø8,169,2ØØ,138
516Ø6 :141,Ø22,2Ø8,169,ØØØ,141,Ø63
51612 :Ø33,2Ø8,173,Ø98,2Ø2,24Ø,Ø86
51618 :Ø54,162,ØØ8,16Ø,Ø14,Ø24,Ø72
51624 :Ø32,24Ø,255,169,Ø42,16Ø,Ø42
5163Ø :2Ø2,Ø32,Ø3Ø,171,173,171,185
51636 :ØØ2,174,17Ø,ØØ2,Ø32,2Ø5,253
51642 :189,162,Ø13,16Ø,Ø13,Ø24,235
51648 :Ø32,24Ø,255,169,Ø92,16Ø,116
51654 :2Ø2,Ø32,Ø3Ø,171,173,169,2Ø7
5166Ø :ØØ2,Ø1Ø,17Ø,189,249,2Ø1,ØØ1
51666 :168,189,248,2Ø1,Ø32,Ø3Ø,Ø54
51672 :171,162,Ø15,173,Ø98,2Ø2,Ø13
51678 :2Ø8,ØØ2,162,Ø11,16Ø,Ø11,ØØ8
51684 :Ø24,Ø32,24Ø,255,169,Ø74,254
5169Ø :16Ø,2Ø2,Ø32,Ø3Ø,171,173,234
51696 :ØØØ,22Ø,Ø74,176,25Ø,Ø76,Ø12
517Ø2 :Ø13,192,ØØ2,2Ø2,Ø1Ø,2Ø2,Ø99
517Ø8 :Ø18,2Ø2,Ø26,2Ø2,Ø34,2Ø2,168
51714 :Ø67,Ø65,Ø8Ø,Ø84,Ø65,Ø73,18Ø
5172Ø :Ø78,ØØØ,Ø32,Ø32,Ø77,Ø65,Ø36
51726 :Ø74,Ø79,Ø82,ØØØ,Ø67,Ø79,139
51732 :Ø76,Ø79,Ø78,Ø69,Ø76,ØØØ,142
51738 :Ø71,Ø69,Ø78,Ø69,Ø82,Ø65,2Ø4
51744 :Ø76,ØØØ,Ø87,Ø65,Ø82,Ø82,168
5175Ø :Ø73,Ø79,Ø82,ØØØ,129,Ø71,216
51756 :Ø65,Ø77,Ø69,Ø32,Ø79,Ø86,196
51762 :Ø69,Ø82,Ø17,Ø17,Ø17,157,153
51768 :157,157,157,157,157,157,23Ø
51774 :157,157,157,158,Ø32,Ø83,Ø38
5178Ø :Ø67,Ø79,Ø82,Ø69,Ø32,ØØØ,141
51786 :154,Ø8Ø,Ø82,Ø69,Ø83,Ø83,113
51792 :Ø32,Ø85,Ø8Ø,Ø32,Ø84,Ø79,216
51798 :Ø32,Ø8Ø,Ø76,Ø65,Ø89,ØØØ,172
518Ø4 :Ø82,Ø65,Ø78,Ø75,Ø32,ØØØ,168
5181Ø :ØØØ,Ø13,Ø13,Ø13,Ø13,Ø13,163
```

Program 3: The Last Warrior, Atari Version

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article before entering this listing.

```
8192:173,148,ØØ2,Ø24,1Ø5,Ø4Ø,236
8198:133,Ø67,173,149,ØØ2,1Ø5,123
82Ø4:ØØØ,133,Ø68,Ø32,Ø17,Ø33,Ø39
821Ø:Ø32,Ø83,Ø36,Ø32,139,Ø33,117
8216:Ø76,ØØ2,Ø39,Ø32,Ø83,Ø33,Ø33
8222:Ø32,144,Ø35,Ø32,21Ø,Ø32,ØØ3
8228:162,ØØØ,169,1ØØ,141,ØØØ,Ø96
8234:ØØ6,141,ØØ5,ØØ6,169,ØØØ,113
824Ø:141,Ø16,ØØ6,2Ø6,Ø22,ØØ6,189
8246:2Ø8,Ø12,173,Ø23,ØØ6,141,1Ø5
8252:Ø22,ØØ6,Ø32,216,Ø33,Ø32,145
8258:223,Ø37,Ø32,227,Ø35,Ø32,14Ø
8264:123,Ø37,169,ØØØ,133,Ø77,Ø99
827Ø:173,252,ØØ2,Ø48,224,169,178
8276:255,141,252,ØØ2,238,25Ø,198
8282:ØØ6,173,252,ØØ2,Ø48,251,Ø54
8288:169,255,141,252,ØØ2,2Ø6,Ø97
8294:25Ø,ØØ6,24Ø,2Ø1,138,Ø72,241
83ØØ:Ø24,1Ø5,Ø44,141,134,Ø32,Ø76
83Ø6:141,191,Ø32,189,2ØØ,ØØ6,1Ø5
8312:141,133,Ø32,169,ØØØ,133,216
8318:2Ø4,141,19Ø,Ø32,16Ø,Ø16,1Ø1
8324:153,255,255,136,2Ø8,25Ø,1Ø9
833Ø:189,Ø16,ØØ6,133,2Ø3,168,Ø85
8336:185,2Ø1,Ø32,157,ØØ8,2Ø8,167
8342:16Ø,ØØ4,ØØ6,2Ø3,Ø38,2Ø4,253
8348:136,2Ø8,249,165,2Ø3,Ø24,117
8354:1Ø5,ØØØ,133,2Ø3,165,2Ø4,2Ø4
836Ø:1Ø5,Ø4Ø,133,2Ø4,189,ØØØ,Ø71
8366:ØØ6,157,ØØØ,2Ø8,189,ØØ5,227
8372:ØØ6,157,2ØØ,ØØ6,17Ø,16Ø,111
8378:ØØØ,177,2Ø3,157,255,255,2Ø9
8384:232,2ØØ,192,Ø16,2Ø8,245,ØØ5
839Ø:1Ø4,17Ø,Ø96,ØØØ,ØØØ,ØØØ,Ø56
8396:ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,2Ø4
84Ø2:169,Ø62,141,Ø47,ØØ2,169,Ø32
84Ø8:ØØ4,141,111,ØØ2,162,ØØ3,127
8414:189,Ø13,Ø33,157,192,ØØ2,Ø4Ø
842Ø:2Ø2,Ø16,247,169,ØØ3,141,238
8426:Ø29,2Ø8,169,Ø4Ø,141,ØØ7,Ø6Ø
8432:212,133,2Ø4,169,ØØØ,133,Ø67
8438:2Ø3,162,ØØ8,16Ø,ØØØ,145,156
8444:2Ø3,136,2Ø8,251,2Ø2,2Ø8,18Ø
845Ø:248,185,Ø44,Ø39,153,ØØØ,159
8456:Ø4Ø,136,2Ø8,247,Ø96,127,Ø94
8462:127,127,127,169,Ø64,133,249
8468:16Ø,162,Ø96,169,Ø12,157,ØØ8
8474:Ø66,ØØ3,Ø32,Ø86,228,162,Ø91
848Ø:Ø96,169,ØØ3,157,Ø66,ØØ3,Ø14
8486:169,Ø99,157,Ø68,ØØ3,169,191
8492:Ø33,157,Ø69,ØØ3,169,ØØ7,226
8498:157,Ø75,ØØ3,169,Ø28,157,127
85Ø4:Ø74,ØØ3,Ø32,Ø86,228,173,14Ø
851Ø:Ø48,ØØ2,133,Ø14,173,Ø49,225
8516:ØØ2,133,Ø15,16Ø,Ø88,169,123
8522:ØØ6,145,Ø14,2ØØ,192,Ø91,21Ø
8528:2Ø8,249,Ø96,16Ø,Ø37,185,247
8534:1Ø1,Ø33,145,Ø67,136,Ø16,Ø72
854Ø:248,169,ØØ7,141,198,ØØ2,Ø89
8546:Ø96,Ø83,Ø58,ØØØ,ØØØ,ØØØ,Ø79
8552:179,163,175,178,165,ØØØ,196
8558:ØØØ,144,144,144,144,144,Ø62
8564:144,144,ØØØ,ØØØ,ØØØ,ØØØ,148
857Ø:ØØØ,ØØØ,Ø51,Ø4Ø,Ø41,Ø37,Ø35
8576:Ø44,Ø36,ØØØ,ØØØ,ØØØ,Ø16,224
8582:Ø21,Ø16,Ø16,Ø16,ØØØ,169,116
8588:ØØ1,141,Ø29,ØØ6,141,Ø43,245
8594:ØØ6,162,Ø53,16Ø,Ø8Ø,169,ØØ8
86ØØ:ØØØ,141,Ø47,ØØ6,169,Ø23,Ø26
86Ø6:Ø32,145,Ø34,162,11Ø,16Ø,Ø33
8612:Ø8Ø,169,ØØØ,141,Ø47,ØØ6,Ø95
8618:169,14Ø,Ø32,145,Ø34,162,Ø84
8624:159,16Ø,Ø5Ø,169,Ø35,141,122
863Ø:Ø47,ØØ6,169,125,Ø32,145,194
8636:Ø34,162,125,16Ø,Ø35,169,1Ø5
8642:Ø34,141,Ø47,ØØ6,169,Ø38,117
8648:Ø32,145,Ø34,162,ØØØ,16Ø,221
8654:Ø5Ø,Ø32,145,Ø34,169,ØØØ,124
866Ø:141,Ø43,ØØ6,Ø96,173,12Ø,Ø23
8666:ØØ2,Ø74,176,Ø1Ø,174,ØØ5,147
8672:ØØ6,224,Ø35,24Ø,ØØ3,2Ø6,17Ø
8678:ØØ5,ØØ6,Ø74,176,Ø1Ø,174,163
8684:ØØ5,ØØ6,224,175,24Ø,ØØ3,121
869Ø:238,ØØ5,ØØ6,Ø74,176,Ø1Ø,239
8696:174,ØØØ,ØØ6,224,Ø53,24Ø,177
87Ø2:ØØ3,2Ø6,ØØØ,ØØ6,Ø74,176,2Ø7
87Ø8:Ø1Ø,174,ØØØ,ØØ6,224,195,1Ø1
8714:24Ø,ØØ3,238,ØØØ,ØØ6,173,158
872Ø:132,ØØ2,2Ø8,ØØ5,162,ØØ1,Ø14
8726:142,Ø21,ØØ6,Ø96,138,Ø72,241
8732:173,Ø29,ØØ6,162,ØØ6,Ø1Ø,158
8738:2Ø2,2Ø8,252,141,Ø48,ØØ6,123
8744:162,ØØ5,Ø32,128,Ø34,133,Ø22
875Ø:186,134,187,162,ØØ3,Ø32,238
8756:128,Ø34,Ø24,1Ø1,186,133,146
8762:186,138,1Ø1,187,133,187,222
8768:1Ø4,Ø74,144,ØØ6,Ø78,Ø48,ØØ6
8774:ØØ6,Ø78,Ø48,ØØ6,Ø74,144,17Ø
878Ø:Ø12,Ø78,Ø48,ØØ6,Ø78,Ø48,Ø9Ø
8786:ØØ6,Ø78,Ø48,ØØ6,Ø78,Ø48,Ø9Ø
8792:ØØ6,Ø24,1Ø1,186,144,ØØ2,Ø39
8798:23Ø,187,Ø24,1Ø1,Ø88,133,Ø89
88Ø4:186,165,187,1Ø1,Ø89,133,193
881Ø:187,16Ø,ØØØ,177,186,174,222
8816:Ø43,ØØ6,2Ø8,ØØ6,Ø77,Ø48,244
8822:ØØ6,Ø76,125,Ø34,Ø13,Ø48,164
8828:ØØ6,145,186,Ø96,169,ØØØ,214
8834:141,Ø26,ØØ6,152,Ø1Ø,Ø46,255
884Ø:Ø26,ØØ6,2Ø2,2Ø8,249,174,233
8846:Ø26,ØØ6,Ø96,141,Ø38,ØØ6,199
8852:Ø72,138,Ø72,152,Ø72,142,Ø28
8858:Ø32,ØØ6,142,Ø35,ØØ6,14Ø,ØØ3
8864:Ø3Ø,ØØ6,14Ø,Ø37,ØØ6,169,Ø36
887Ø:ØØØ,141,Ø34,ØØ6,141,Ø36,Ø12
8876:ØØ6,141,Ø46,ØØ6,173,Ø47,Ø79
8882:ØØ6,141,Ø39,ØØ6,173,Ø38,Ø69
8888:ØØ6,2Ø5,Ø32,ØØ6,176,Ø18,115
8894:169,ØØ1,141,128,ØØ6,173,Ø4Ø
89ØØ:Ø32,ØØ6,Ø56,237,Ø38,ØØ6,Ø59
89Ø6:141,168,ØØ6,Ø76,223,Ø34,Ø82
8912:169,ØØØ,141,128,ØØ6,173,Ø57
8918:Ø38,ØØ6,Ø56,237,Ø32,ØØ6,Ø77
8924:141,168,ØØ6,173,Ø3Ø,ØØ6,232
893Ø:Ø56,237,Ø39,ØØ6,141,176,113
8936:ØØ6,173,Ø35,ØØ6,17Ø,173,Ø27
8942:Ø37,ØØ6,172,Ø43,ØØ6,2Ø8,198
8948:Ø16,2Ø5,Ø44,ØØ6,2Ø8,ØØ5,216
8954:236,Ø45,ØØ6,24Ø,Ø1Ø,141,16Ø
896Ø:Ø44,ØØ6,142,Ø45,ØØ6,168,155
8966:Ø32,Ø26,Ø34,173,128,ØØ6,149
8972:2Ø8,Ø21,173,Ø34,ØØ6,Ø24,222
8978:1Ø9,168,ØØ6,141,Ø34,ØØ6,226
8984:173,Ø35,ØØ6,1Ø5,ØØØ,141,228
899Ø:Ø35,ØØ6,Ø76,Ø53,Ø35,173,152
8996:Ø34,ØØ6,Ø56,237,168,ØØ6,Ø31
9ØØ2:141,Ø34,ØØ6,173,Ø35,ØØ6,181
9ØØ8:233,ØØØ,141,Ø35,ØØ6,173,124
9Ø14:Ø36,ØØ6,Ø56,237,176,ØØ6,Ø59
9Ø2Ø:141,Ø36,ØØ6,173,Ø37,ØØ6,2Ø3
9Ø26:233,ØØØ,141,Ø37,ØØ6,2Ø6,177
9Ø32:Ø46,ØØ6,2Ø8,157,1Ø4,168,249
9Ø38:1Ø4,17Ø,1Ø4,Ø96,162,ØØ3,2Ø5
9Ø44:Ø32,1Ø6,Ø32,2Ø2,Ø16,25Ø,21Ø
9Ø5Ø:Ø96,216,173,25Ø,ØØ6,2Ø8,Ø15
9Ø56:ØØ9,Ø32,Ø82,Ø35,Ø32,115,145
9Ø62:Ø36,Ø32,1Ø9,Ø35,Ø76,Ø98,232
9Ø68:228,238,246,ØØ6,173,246,221
9Ø74:ØØ6,141,ØØ4,21Ø,2Ø1,Ø45,2Ø9
```



Notice how distant aliens appear smaller and nearby ships loom larger in the Atari version of "The Last Warrior."

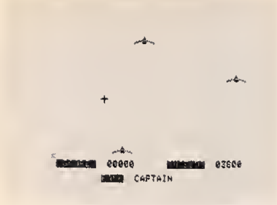
```
9080:208,005,169,020,141,246,141
9006:006,173,247,006,240,011,041
9092:206,247,006,074,074,074,045
9098:041,239,141,007,210,096,104
9104:160,091,162,035,169,007,000
9110:032,092,220,169,200,141,244
9116:023,006,169,000,141,021,004
9122:006,141,040,006,141,249,233
9128:006,141,250,006,162,003,224
9134:032,049,037,169,000,157,106
9140:200,006,141,030,200,202,207
9146:208,242,169,000,133,119,033
9152:169,040,133,120,162,008,056
9158:160,000,152,145,119,136,142
9164:208,251,230,120,202,200,143
9170:246,169,003,141,015,210,226
9176:169,004,141,000,210,169,149
9182:255,141,252,002,096,173,117
9100:021,006,240,044,169,025,221
9194:141,000,210,141,246,006,210
9200:169,100,141,001,210,206,043
9206:021,006,169,002,032,026,246
9212:036,169,002,032,055,036,070
9218:169,002,032,026,036,169,180
9224:002,032,055,036,169,002,048
9230:141,240,006,032,065,036,030
9236:169,000,141,001,210,096,125
9242:141,029,006,162,000,160,012
9248:079,173,000,006,056,233,067
9254:045,072,173,005,006,056,139
9260:233,028,074,141,047,006,061
9266:104,032,145,034,096,141,090
9272:029,006,162,159,160,079,139
9278:076,033,036,173,249,006,123
9204:200,012,160,035,162,003,136
9290:032,086,030,206,240,006,178
9296:200,239,096,165,000,133,241
9302:203,165,089,133,204,162,010
9308:012,160,000,173,010,210,145
9314:201,020,176,004,041,003,031
9320:145,203,200,208,242,230,052
9326:204,202,208,237,096,162,195
9332:003,189,208,006,240,003,253
9338:076,042,037,189,128,006,088
9344:208,027,189,149,006,024,219
9350:125,168,006,157,149,006,233
9356:189,000,006,105,000,157,085
9362:000,006,201,205,144,030,220
9368:032,049,037,144,025,189,116
9374:149,006,056,253,168,006,028
9380:157,149,006,189,000,006,159
9386:233,000,157,000,006,201,255
9392:015,176,003,032,049,037,232
9398:189,144,006,208,027,189,177
9404:154,006,024,125,176,006,167
9410:157,154,006,189,005,006,199
9416:105,000,157,005,006,201,162
9422:186,144,030,032,049,037,172
9428:144,025,189,154,006,056,018
9434:253,176,006,157,154,006,202
9440:189,005,006,233,000,157,046
9446:005,006,201,020,176,003,129
9452:032,049,037,222,160,006,230
9458:208,054,169,020,157,160,242
9464:006,189,016,006,201,006,160
9470:144,039,173,040,006,208,096
9476:037,173,010,210,201,080,203
9402:176,030,169,001,141,040,055
9488:006,109,000,006,056,233,250
9494:045,141,240,006,189,005,136
9500:006,056,233,020,074,141,054
9506:241,006,076,042,037,254,170
9512:016,006,202,240,003,076,071
9518:117,036,096,173,010,210,176
9524:157,000,006,173,010,210,096
9530:201,100,176,249,157,005,002
9536:006,160,000,173,010,210,111
9542:016,002,160,001,152,157,046
9540:120,006,160,000,173,010,041
9554:210,016,002,160,001,152,111
9560:157,144,006,173,010,210,020
9566:074,024,105,050,157,176,160
9572:006,173,010,210,074,024,085
9578:105,050,157,168,006,169,249
9584:100,157,160,006,169,001,193
9590:157,016,006,024,096,173,070
9596:040,006,240,083,174,240,139
9602:006,240,073,224,150,176,239
9608:069,172,241,006,192,021,069
9614:144,062,192,070,176,058,004
9620:152,056,233,020,141,047,029
9626:006,173,240,006,109,010,186
9632:210,201,105,176,246,201,099
9638:055,144,242,169,003,141,152
9644:029,006,032,145,034,072,234
9650:169,055,141,200,002,104,081
9656:032,145,034,032,212,037,164
9662:169,000,141,200,002,032,222
9660:074,030,173,249,006,240,200
9674:003,076,126,030,169,000,102
9600:141,040,006,096,162,007,140
9686:160,000,136,200,253,202,149
9692:208,250,096,162,003,109,104
9698:200,006,240,025,222,208,111
9704:006,208,030,032,049,037,002
9710:238,250,006,032,106,032,134
9716:169,000,141,250,006,141,103
9722:030,200,076,009,038,189,032
9720:012,208,041,001,240,003,249
9734:076,013,030,202,200,213,244
9740:096,169,000,141,030,200,144
9746:173,021,006,240,242,032,220
9752:040,038,169,007,157,016,203
9750:006,169,003,157,200,006,067
9764:169,120,141,247,006,169,120
9770:150,141,006,210,200,217,206
9776:160,014,130,072,162,004,006
9702:056,177,067,105,000,201,140
9788:154,144,002,169,144,145,050
9794:067,136,202,016,240,104,063
9800:170,096,160,034,162,003,105
9806:169,010,141,240,006,076,216
9812:065,036,024,177,067,233,174
9810:000,201,015,240,005,056,095
9824:145,067,176,005,169,025,171
9830:145,067,024,136,202,016,100
9836:234,160,035,177,067,201,214
9842:016,208,008,136,192,031,193
9848:208,245,238,249,006,096,138
9854:162,003,169,002,157,016,123
9860:006,202,208,250,169,255,198
9866:141,247,006,169,150,141,224
9872:006,210,173,010,210,141,126
9878:200,002,041,007,170,189,247
9884:001,040,160,000,145,014,004
9890:173,247,006,208,235,169,176
9896:112,145,014,169,000,141,237
9902:200,002,160,022,185,150,125
9908:039,145,067,200,192,038,093
9914:208,246,032,030,039,160,133
9920:024,185,164,039,145,067,048
9926:200,192,029,208,246,160,209
9932:011,177,067,201,144,240,020
9938:004,169,008,208,012,200,043
9944:192,013,208,241,177,067,090
9950:056,233,144,074,010,170,141
9956:189,209,039,141,245,038,065
9962:189,210,039,141,246,038,073
9968:162,000,160,029,189,255,011
9974:255,240,006,145,067,200,135
9980:232,208,245,032,030,039,014
9986:160,022,185,171,039,145,212
9992:067,200,192,038,208,246,191
9998:173,132,002,208,251,173,185
10004:132,002,240,251,162,255,038
10010:154,076,027,032,169,255,227
10016:141,248,006,032,212,037,196
10022:206,248,006,200,248,096,026
10028:000,016,016,016,124,124,004
10034:016,016,016,000,000,000,098
10040:000,000,000,000,000,000,056
10046:000,000,000,000,000,000,062
10052:000,000,000,000,000,000,076
10050:000,000,000,000,000,000,074
10064:000,000,000,000,012,000,092
10070:000,000,000,000,000,000,086
10076:000,000,000,000,000,000,092
10082:000,008,028,000,000,000,134
10088:000,000,000,000,000,000,104
10094:000,000,000,000,000,000,118
10100:028,020,000,000,000,000,172
10106:000,000,000,000,000,000,122
10112:000,000,008,000,020,034,206
10110:000,000,000,000,000,000,134
10124:000,000,000,000,000,000,148
10130:008,020,062,085,000,000,073
10136:000,000,000,000,000,000,152
10142:000,040,060,048,134,065,001
10148:148,066,148,066,036,020,136
10154:000,000,000,000,000,039,209
10160:033,045,037,000,047,054,136
10166:037,050,000,000,000,000,013
10172:050,033,046,043,000,048,152
10178:050,037,051,051,000,038,165
10104:041,050,037,034,053,052,211
10190:052,047,046,219,039,227,060
10196:039,233,039,241,039,249,020
10202:039,035,033,048,052,033,202
10200:041,046,000,045,033,042,175
10214:047,050,000,035,047,044,197
10220:047,046,037,044,000,039,193
10226:037,046,037,050,033,044,233
10232:000,055,033,050,050,041,221
10238:047,050,000,000,016,032,143
10244:040,064,000,096,112,000,148
```

Program 4: The Last Warrior, Apple Version

Version by Tim Victor, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
CF 100 D$ = CHR$ (4): DIM P$(B),
       PX(3),PY(3),PZ(3),VX(3),V
       Y(3),QX(3),QY(3),QZ(3),R$
       (4)
BF 110 GOSUB 1000
3E 120 SH = 5000:SC = 0
Ø3 130 P$(0) = "/012":P$(1) = "(
       )*+":P$(2) = "!" + CHR$ (
       34) + "#$":P$(3) = "34":P
       $(4) = "%&":P$(5) = ",-"
77 140 P$(6) = "5":P$(7) = ".":P
       $(B) = "'"
5B 150 GOSUB 970
C4 160 FOR I = 0 TO 3:PZ(I) = 10
       00:QZ(I) = 1000: NEXT
7D 170 CO = .95:SH = 5000:SC = 0
       : GOBUB 910: GOSUB 930
C2 1B0 XP = 52:YP = 59: XDRAW 1
       AT XP,YP
ØF 190 RF = 0: FOR M = 0 TO 3
71 200 IF SH = 0 THEN 330
9C 210 I = FRE (0): GOSUB 400: O
       N I GOSUB 430,440,450,460
       ,470,560
5F 220 IF PZ(M) = 1000 THEN GOSU
       B 570: GOTO 310
FC 230 IF PZ(M) < 0 THEN RF = 1:
        GOTO 300
4B 240 GOSUB 610
F3 250 IF PZ(M) > 15 THEN 300
7A 260 IF RND (1) < CO * .B THEN
        300
```



The Apple version of "The Last Warrior" animates the alien ships using custom characters designed with the previously published "Apple SuperFont" utility.

```
Ø1 27Ø XDRAW 1 AT XP,YP:XT = INT
        (PX(M)) * 7 - 7:YT = INT
        (PY(M)) * 8 - 4
ØE 28Ø HCOLOR= 5: GOSUB 37Ø: HCD
       LOR= Ø: GOSUB 37Ø: GOSUB
       65Ø: XORAW 1 AT XP,YP
E8 29Ø SH = SH - 1ØØ: GDSUB 93Ø
F3 3ØØ CO = CO * .9999: NEXT
92 31Ø IF RF < > Ø THEN XDRAW 1
       AT XP,YP: GOSUB 65Ø: XORA
       W 1 AT XP,YP
99 32Ø GOTD 19Ø
64 33Ø XORAW 1 AT XP,YP: VTAB 21
       : HTAB 2: PRINT "ANOTHER
       GAME? (Y OR N)"
46 34Ø GET A$: IF A$ = "N" OR A$
        = "n" THEN TEXT : END
5A 35Ø IF A$ = "Y" OR A$ = "y" T
       HEN 15Ø
9D 36Ø GOTO 34Ø
9A 37Ø HPLDT XT,YT TD Ø,Ø: HPLDT
        XT,YT TO Ø,159
85 38Ø HPLOT XT,YT TO 279,Ø: HPL
       OT XT,YT TO 279,159
25 39Ø RETURN
5E 4ØØ I = Ø:A = PEEK (49152)
B4 41Ø IF A > 127 THEN PDKE 4916
       8,Ø:A$ = CHR$ (A - 128):
       FOR I = 1 TO 6: IF A$ < >
        MIO$ ("JILK P",I,1) THEN
        NEXT
18 42Ø RETURN
6E 43Ø GOSUB B9Ø:XP = XP - (XP >
        6) * 7: GOTO 9ØØ
48 44Ø GDSUB B9Ø:YP = YP - (YP >
        7) * 8: GOTO 9ØØ
74 45Ø GDSUB 89Ø:XP = XP + (XP <
        273) * 7: GOTO 9ØØ
D4 46Ø GOSUB 89Ø:YP = YP + (YP <
        152) * 8: GOTD 9ØØ
CA 47Ø HCDLDR= 7: XDRAW 1 AT XP,
       YP: HPLOT Ø,159 TO XP,YP:
        HPLOT 279,159 TO XP,YP
E1 48Ø HCOLOR= Ø: HPLOT Ø,159 TO
        XP,YP: HPLOT 279,159 TO
       XP,YP
57 49Ø XC = INT (XP / 7) + 1:YC
       = INT (YP / 8) + 1: FOR J
        = Ø TD 3: IF PZ(J) = 1ØØ
       Ø THEN 54Ø
1B 5ØØ IF YC < > INT (PY(J)) THE
       N 54Ø
E8 51Ø IF XC < INT (PX(J)) - (PZ
       (J) < = 15) - (PZ(J) < =
       3Ø) THEN 54Ø
61 52Ø IF XC > INT (PX(J)) + (PZ
       (J) < = 15) THEN 54Ø
4D 53Ø PZ(J) = - PZ(J): GOSUB 65
       Ø:SC = SC + 1ØØ: GOSUB 91
       Ø: GOTO 55Ø
Ø5 54Ø NEXT
91 55Ø XDRAW 1 AT XP,YP:SH = SH
       - 2Ø: GOSUB 93Ø: RETURN
67 56Ø GET A$: RETURN
4A 57Ø IF RND (1) < CO THEN 6ØØ
B8 58Ø PX(M) = RNO (1) * 35 + 3:
       PY(M) = RNO (1) * 2Ø + 1:
       PZ(M) = 45
F7 59Ø R = RNO (1) - .5:VX(M) =
       ( ABS (R) - .25):VY(M) =
       SQR (.Ø625 - VX(M) * VX(M
       )) * SGN (R):RF = 1
16 6ØØ RETURN
85 61Ø PX(M) = PX(M) + VX(M) * (
       PX(M) > 4 ANO PX(M) < 37)
       : IF INT (QX(M)) < > INT
       (PX(M)) THEN RF = 1
5D 62Ø PY(M) = PY(M) + VY(M) * (
       PY(M) > 2 ANO PY(M) < 2Ø)
       : IF INT (QY(M)) < > INT
       (PY(M)) THEN RF = 1
77 63Ø PZ(M) = PZ(M) - 2 * (PZ(M
       ) > 2): IF PZ(M) = 3Ø OR
       PZ(M) = 15 THEN RF = 1
1E 64Ø RETURN
86 65Ø FOR I = Ø TO 3: IF QZ(I)
       = 1ØØØ THEN 73Ø
4A 66Ø NF = QZ(I):QZ(I) = ABS (Q
       Z(I))
ØD 67Ø IF QZ(I) < = 15 THEN GOSU
       B BØØ: GOTO 7ØØ
8F 68Ø IF QZ(I) < = 3Ø THEN GOSU
       B B1Ø: GOTO 7ØØ
5A 69Ø GOSUB 82Ø
E8 7ØØ IF NF > = Ø THEN 73Ø
Ø8 71Ø IF I < 3 THEN GOSUB 87Ø:I
        = I - 1
67 72Ø QZ(3) = 1ØØØ
3D 73Ø NEXT : FOR I = 3 TO Ø STE
       P - 1: IF PZ(I) = 1ØØØ TH
       EN 78Ø
8Ø 74Ø QX(I) = PX(I):QY(I) = PY(
       I):QZ(I) = PZ(I)
EC 75Ø IF ABS (PZ(I)) < = 15 THE
       N GO8U8 83Ø: GOTO 78Ø
FE 76Ø IF ABS (PZ(I)) < = 3Ø THE
       N GOSUB 84Ø: GOTO 78Ø
5D 77Ø GDSUB 85Ø
ØF 78Ø NEXT
29 79Ø RETURN
19 8ØØ HTAB QX(I) - 2: VTAB QY(I
       ): PRINT "     ": RETURN
FA 81Ø HTAB QX(I) - 1: VTAB QY(I
       ): PRINT "   ": RETURN
A6 B2Ø HTAB QX(I): VTAB QY(I): P
       RINT " ": RETURN
42 83Ø GOSUB 86Ø: HTAB PX(I) - 2
       : VTAB PY(I): PRINT P$(PH
       ): RETURN
98 B4Ø GOSUB 86Ø: HTAB PX(I) - 1
       : VTAB PY(I): PRINT P$(PH
        + 3): RETURN
21 85Ø GOSUB 86Ø: HTAB PX(I): VT
       AB PY(I): PRINT P$(PH + 6
       ): RETURN
FD 86Ø PH = (PZ(I) > = Ø) * INT
       (PX(I) - 2 * INT (PX(I) /
        2) + 1): RETURN
38 87Ø FOR K = I TO 2:PX(K) = PX
       (K + 1):PY(K) = PY(K + 1)
       :PZ(K) = PZ(K + 1)
55 88Ø VX(K) = VX(K + 1):VY(K) =
        VY(K + 1):QX(K) = QX(K +
        1):QY(K) = QY(K + 1):QZ(
       K) = QZ(K + 1): NEXT :PZ(
       3) = 1ØØØ: RETURN
1B 89Ø OX = XP:DY = YP: RETURN
76 9ØØ XDRAW 1 AT DX,DY: XORAW 1
        AT XP,YP: RETURN
1A 91Ø N$ = STR$ (SC): VTAB 22:
       HTAB 11: GDSUB 95Ø:R = IN
       T (SC / 2ØØØ): IF R > 4 T
       HEN R = 4
AC 92Ø VTAB 24: HTAB 16: CALL -
       868: PRINT R$(R);: RETURN
7E 93Ø IF SH < Ø THEN SH = Ø
55 94Ø N$ = STR$ (SH): VTAB 22:
       HTAB 31: GOTD 95Ø
3E 95Ø IF LEN (N$) < 5 THEN PRIN
       T LEFT$ ("ØØØØ",5 - LEN (
       N$));
48 96Ø PRINT N$;: RETURN
Ø7 97Ø HOME : HGR : INVERSE : VT
       AB 22: HTAB 2: PRINT " SC
       ORE ";: HTAB 22: PRINT "S
       HIELDS";
5Ø 98Ø VTAB 24: HTAB 1Ø: PRINT "
       RANK";
F1 99Ø NORMAL : RETURN
A3 1ØØØ POKE 232,1ØØ: PDKE 233,3
2Ø 1Ø1Ø POKE B68,1: POKE 87Ø,4:
        POKE 871,Ø
48 1Ø2Ø FDR I = Ø TD 4: READ A:
        POKE 872 + I,A: NEXT
AF 1Ø3Ø HCOLOR= 7: RDT= Ø: SCALE
        = 4
42 1Ø4Ø FOR I = Ø TD 4: READ R$(
        I): NEXT
3C 1Ø5Ø FOR I = 76B TD I + B7: R
        EAO A: POKE I,A: NEXT
54 1Ø6Ø FOR I = 13B * 256 TO I +
         175: REAO A: POKE I,A:
        NEXT
A6 1Ø7Ø IF PEEK (191 * 256) = 76
         THEN PRINT O$;"PR#A$3ØØ
        ": GOTO 1Ø9Ø
35 1Ø8Ø POKE 54,Ø: POKE 55,3: CA
        LL 1ØØ2
14 1Ø9Ø PDKE 6,Ø: POKE 7,138: RE
        TURN
22 11ØØ OATA 176,12,31,5,Ø
68 111Ø OATA CAPTAIN,MAJOR,COLON
        EL,GENERAL,WARRIOR
D6 112Ø OATA 216,12Ø,133,69,134,
        7Ø
2E 113Ø OATA 132,71,166,7,1Ø,1Ø
44 114Ø DATA 176,4,16,62,48,4
88 115Ø DATA 16,1,232,232,1Ø,134
6Ø 116Ø OATA 27,24,1Ø1,6,133,26
A3 117Ø OATA 144,2,23Ø,27,165,4Ø
95 118Ø OATA 133,8,165,41,41,3
81 119Ø OATA 5,23Ø,133,9,162,B
18 12ØØ OATA 16Ø,Ø,177,26,36,5Ø
63 121Ø OATA 48,2,73,127,164,36
47 122Ø DATA 145,8,23Ø,26,2Ø8,2
9F 123Ø DATA 23Ø,27,165,9,24,1Ø5
ØF 124Ø OATA 4,133,9,2Ø2,2Ø8,226
87 125Ø OATA 165,69,166,7Ø,164,7
        1
72 126Ø OATA 88,76,24Ø,253
6A 127Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
E8 128Ø OATA Ø,Ø,Ø,Ø,Ø,4Ø,42,2
5D 129Ø OATA 64,64,96,16,21,117,
        112,Ø
19 13ØØ OATA Ø,Ø,1,2,42,43,3,Ø
87 131Ø OATA Ø,Ø,Ø,Ø,Ø,5,21,16
84 132Ø OATA Ø,Ø,Ø,Ø,64,1Ø4,66,Ø
A9 133Ø DATA Ø,Ø,Ø,1,3,23,67,Ø
4Ø 134Ø OATA Ø,Ø,Ø,Ø,8,42,Ø,Ø
74 135Ø OATA Ø,Ø,Ø,Ø,64,84,21,1
2A 136Ø OATA 32,32,112,8,1Ø,122,
        12Ø,Ø
C3 137Ø OATA Ø,Ø,Ø,1,21,85,65,Ø
64 138Ø OATA Ø,Ø,Ø,Ø,Ø,2,1Ø,B
12 139Ø OATA Ø,Ø,Ø,64,96,116,97,
        Ø
E7 14ØØ OATA Ø,Ø,Ø,Ø,1,11,33,Ø
22 141Ø OATA Ø,Ø,Ø,Ø,4,21,Ø,Ø
45 142Ø DATA 24,48,24,64,118,3,5
        6,Ø
9D 143Ø DATA 56,99,48,55,88,111,
        1Ø2.Ø
A4 144Ø OATA 6,12,63,27,113,31,1
        12,Ø
75 145Ø DATA 3,6,48,1Ø8,12,51,Ø,
        Ø
BB 146Ø DATA Ø,56,99,48,55,88,Ø,
        Ø
7F 147Ø OATA Ø,6,12,63,27,113,Ø,
        Ø
5A 148Ø DATA Ø,Ø,76,118,54,Ø,Ø,Ø
```

©

Rescue On Fractalus! And *Ballblazer*

Tom R. Halfhill, Editor

Requirements: Atari 400/800, XL, or XE computer with at least 48K RAM, a disk drive, and a joystick (two joysticks are recommended for Ballblazer*). Versions for the Commodore 64 and Apple II-series computers were due to be released early this summer (except for the 64 version of* Ballblazer*, which is still under development).*

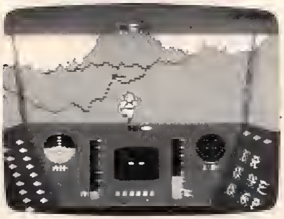
Delayed for a frustrating year by the turmoil of the home computer wars, *Rescue on Fractalus!* and *Ballblazer* have finally hit the market for Atari computers and are pending for the Commodore 64 and Apple as well. It's about time, too, because these action games have been anxiously awaited since their unveiling in mid-1984. Designed by Lucasfilm—the production company which brought us the *Star Wars* trilogy—both games were supposed to be marketed in cooperation with Atari. Unfortunately, Atari fell on hard times and the Lucasfilm games fell into limbo.

For a while, enthusiasts wondered if the games would ever see the glow of home video screens. Tantalizing preproduction copies of *Ballblazer* were known to be circulating in the pirate underground. Finally, Epyx, Inc. clinched a deal with Lucasfilm to market the programs. Now everyone can decide: Were they worth the wait?

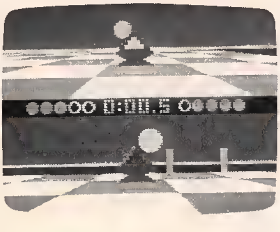
A Mission Of Mercy

Rescue on Fractalus! integrates the best features of Brøderbund's *Choplifter*, Atari's *Star Raiders*, and Microprose's *Solo Flight*. Similar to *Choplifter*, your mission is to locate and rescue fellow pilots stranded in enemy territory—while fighting off hostile aircraft and ground targets. As in *Star Raiders*, you fly a spaceship from a first-person perspective—the video screen is a windshield onto the world beyond. And like *Solo Flight*, success depends on your ability to skillfully maneuver over an ever-changing landscape—while keeping an eye on your flight instruments at the bottom of the screen.

The scenario is that a number of space pilots have been shot down by alien Jaggies on the planet Fractalus. (The planetary landscape is generated by fractal mathematics—get it?) You're an old-fashioned air pilot who has been called back into the Ethercorps to rescue the downed space pilots. Launched from an orbiting mother ship, you have to save a certain quota of pilots during each mission to advance to the next level. The task involves locating the pilots one by one, landing within walking distance, waiting for the pilot to enter your airlock, and then taking off again to resume the search. When your quota is filled, you return the pilots to the mother ship. Meanwhile, you have to duel with Jaggi gun emplacements dug into the mountainsides and fight off kamikaze attacks by Jaggi saucers.



Rescue on Fractalus!: *As you look out onto the jagged mountains of Fractalus, a downed space pilot runs for the safety of your airlock.*



Ballblazer: *With only a half-second left to play and the score 4-3, player two (bottom window) tries to shove the Plasmorb past player one (top window) and into the goal.*

Your craft, a modified Valkyrie-class fighter, is equipped with defense shields, Antimatter Bubble Torpedos, a targeting scope, a long-range scanner that picks up the presence of nearby space pilots, and a detector that warns when a Jaggi gun has locked onto your ship. Flight instruments include an artificial horizon, an energy-level meter, two altimeters, a compass, a speed indicator, a device that shows the clearance between your wingtips and the canyon walls, and digital readouts that tell how many Jaggies you've destroyed, how many pilots you have to rescue, and your distance from the pilot on the long-range scanner. All these dials and gauges are especially important on the highest levels, because you have to fly at night on instruments only.

A team of eight people created this game, and the attention to detail shows. In fact, the flight simulation could be a game in itself. You can climb, dive, and bank by steering the sensitive joystick, and keyboard controls let you speed up, slow down, land, switch your shields on and off, and open the airlock doors. Sound effects are rich: the whine of your engines, the explosions of torpedos and Jaggi gunshots, the anxious knock of pilots pounding on your airlock door to be rescued, and the hiss of the door as it opens and closes. Even the documentation is entertaining and professionally done.

Rescue on Fractalus!, like *Star Raiders*, calls for strategic thinking and contains some surprises and secrets for you to discover before you can move to the highest levels. It's definitely not a fast-paced twitch game. Indeed, at times it moves rather slowly as you search for the stranded pilots. But overall, it's an exceptional effort.

Split-Screen Soccer

Lucasfilm's other release, *Ballblazer*, is equally impressive. The split-screen, high-speed graphics of this frenetic game must be seen to be believed. Like *Rescue on Fractalus!*, it's a first-person perspective game that shows you the view from the driver's seat. But *Ball-*

blazer goes a step further and actually splits the screen into *two* views—one for each player. Two people can compete using two joysticks, or one person can play the computer.

Essentially, *Ballblazer* is space-age soccer played on a checkered field that measures 21 squares wide by 55 squares long (each square represents 5 × 5 meters). The Grid, as it's known, has a pair of goalposts at each end and is surrounded by force fields to keep players from straying out of bounds. As in soccer, the object is to score more goals than your opponent.

Unlike old-fashioned soccer, however, this game isn't played by teams of flesh-and-blood athletes trying to kick around a rubber ball. Instead, there are only two players, and each one drives a fast-moving hovercraft called a Rotofoil. The "ball" is a Plasmorb, a glowing object that floats two meters above the playing field. When you push the joystick forward to cruise over the Grid, your Rotofoil automatically points itself toward the Plasmorb. When you make contact, a force field grabs the Plasmorb and locks it in front of your Rotofoil. Then the Rotofoil reorients itself toward your goal, and away you go.

If you shove the Plasmorb between the goalposts, you get one point. By pressing the joystick button, you can also shoot the Plasmorb forward, recoiling your Rotofoil backward. By shooting the Plasmorb through the goal at close or intermediate range, you can score one or two points. You can even get three points by scoring a goal with an over-the-horizon shot (since the Grid is slightly curved, the goalposts are invisible at long range).

Meanwhile, of course, your computer or human opponent pursues in another Rotofoil, trying to block your shots and steal the Plasmorb. Whoever scores the most goals before the clock expires—usually three minutes—is the winner.

Like most sports, *Ballblazer* appears simple but actually contains many hidden strategies and possibilities. Championship play requires good defensive as well as offensive tactics. You can develop these skills by playing practice games against the computer (with adjustable difficulty levels) and by studying the amusing manual. *Ballblazer* looks like a three-point goal for Lucasfilm and Epyx.

Rescue on Fractalus!
Ballblazer
Epyx, Inc.
1043 Kiel Court
Sunnyvale, CA 94089
$40 each

Below The Root

Nick Piazza, Jr.

Requirements: Commodore 64 with a disk drive; Apple II-series computer with at least 48K RAM and a disk drive; IBM PC with at least 64K RAM, a disk drive, and color/graphics adapter; or an Enhanced Model IBM PCjr. A joystick is required for the 64 version and recommended for the Apple and IBM versions.

It didn't take long for Hollywood to realize that great books could often be made into great movies. The software industry appears to have made the same discovery, and Windham Classics has developed a superb adaptation of Zilpha Keatley Snyder's *Green Sky Trilogy*. (In fact, Snyder collaborated with programmer Dale Disharoon to create *Below the Root*.)

The *Green Sky Trilogy* is set in a fantasy world of trees and tunnels known as Green Sky, and it's up to a character on a quest to save this world from pending destruction. *Below the Root* casts the player as the quester in an enchanting blend of an action and adventure game. It has been designed for players aged ten to adult, but my seven-year-old daughter was able to enjoy the game while playing with a grownup. It's even more enjoyable when several people join together to guide the quest. Indeed, one of the game's strong points is that it encourages cooperation rather than isolated play or deadly competition.

Colorful Graphics

One of the first things that impresses you about *Below the Root* is the quality of the screen graphics—the color and detail rival that of any arcade game. There are more than 100 different screens, each a delight to the eye.

Unlike text adventures, *Below the Root* doesn't require you to enter your commands by typing short sentences such as "Look North" or "Take Object." Instead, you select functions from various menus of choices (with the joystick, if you're using one). This makes the game more suitable for younger children. For example, the main menu lets you start a new game, save a current game on disk, continue a previously saved game, or view a sample game simply by indicating your choice. The last option, by the way, is particularly recommended for first-time players—it's wise to take a few minutes to orient yourself before plunging headlong into this unknown world.

After reading the well-written instructions and viewing the sample game, you're ready to start. First, the program asks which of five questers you wish to adopt. Each comes with varying degrees of stamina and "spirit skill." Questers also represent the two races which occupy Green Sky: the tree-loving Kindar and their cousins, the Erdling. Each race has its own attributes and limitations. All the questers, however, can grow in strength and spirit as they progress through the game.

What really sets this game apart is that questers can be either male or female. My daughter thought it was unfair that she was limited to choosing between three male characters and only two female characters, but still, at a time when computers are becoming increasingly important, it's gratifying to find a game that goes out of its way to encourage young girls as well as boys.

The level of each quester's spirit skill is an important factor in mastering the environment of Green Sky and successfully completing the quest. Spirit skills include the ability to read the emotions and thoughts of others (*pensing*), to heal yourself if injured, to influence tree growth (*grunspreke*), or to move yourself or other objects with your mind (*kiniport*). Each requires higher levels of spirit skill, and it's up tc the player to determine how to raise this level. Those new to Green Sky should select questers with more spirit skill, while those who have played before may want to try questers with less spirit skill for a more challenging game.

Once you've selected your quester, the game begins in the quester's home. At this point, you have 50 days (in game time) to complete your quest and save Green Sky. Initial supplies are available in the quester's home, and players decide their course of action by making selections from the options menu. Many of these options are familiar to those who have played text adventures. You can examine, take, buy, eat, offer, drop, or sell various objects. You can also list an inventory of what you're carrying and call upon your spirit skills.

Quester, Heal Thyself

Questers are free to move throughout Green Sky in various ways: They can walk, run, jump, glide, climb, crawl, or enter and exit buildings. Since much of the action occurs in the treetops of Green Sky, you must be careful not to fall—unless you have a *shuba* for gliding, your quester will suffer a bump on the head. But watching the comical way in which questers rub their heads after a fall may help soothe the pain.

When you first encounter other characters in the game, an important spirit skill to use is pensing. This allows you to determine if they're friendly before speaking to them. This is vital, because some inhabitants are hostile. From time to time, it's also important to check your status, get adequate rest, eat when you're hungry, and heal yourself of any injuries. If your situation becomes too desperate, you may have to *renew* yourself. This option returns you home, but costs you a day from your quest.

The renew option, incidentally, spotlights another attractive feature of *Below the Root*: Questers are never killed or destroyed during their quest. While the world may be lost, violence rarely befalls the quester. This may be an important consideration for young players who would become upset if a character they created was destroyed during a game, or for parents who are disturbed by violence in computer games.

Below the Root
Windham Classics/Spinnaker Software
One Kendall Square
Cambridge, MA 02139
$26.95

Companion

Roger B. Crampton

Requirements: TI-99/4A with 32K RAM expansion card or box, Extended BASIC, a disk drive, and a printer.

Until I saw *Companion*, I considered replacing my TI-99/4A with a much more expensive computer for my serious word processing needs. I had tried several other word processors and found them either too slow, too cumbersome, or lacking essential features. But *Companion*, an inexpensive program written entirely in machine language, solves all of those problems.

Companion's editing features are superb—you have instantaneous full-screen editing capability. And the editing comes naturally, because all normal features of the TI keyboard retain their functions. For example, pressing Function 2 (Insert) works the same way with *Companion* as it does when you're entering a program in console or Extended BASIC. There are no surprises or tricky key sequences with *Companion*. Everything is logical and works in much the same manner as screen editing in BASIC. A delightful exception is the up- and down-arrow keys—they really move the cursor up and down, the way you wish they did in BASIC.

Of course, *Companion* has all of the usual word processing features. You can center headings, set tabs, automatically indent new paragraphs, search for text strings, and move or copy blocks of text. And you don't have to memorize a complex series of keystrokes to do simple things. For instance, pressing CTRL-P automatically generates a linefeed, a carriage return, and indents five spaces for the next paragraph.

The manual is well-written, succinct, and most important, understandable. At 142 pages, it may seem intimidating at first, but there is a good reason for its length. *Companion* has so many features that it takes that many pages to describe them.

Companion works flexibly with different kinds of printers. It lets you send control characters so you can switch to compressed or expanded fonts, or any other fonts allowed by your printer. A little judicious study of your printer manual, along with the *Companion* manual, should enable you to produce a brief list of control characters to adjust nearly any printer parameter.

Companion
Intelpro
5825 Baillargeon Street
Brossard, Quebec
Canada J42 1T1
$79.95

Jr-Draw For PCjr

Norm Cohen

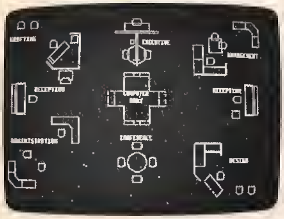
Requirements: Enhanced Model IBM PCjr. Light pen optional.

Jr-Draw is an interactive program which allows a PCjr user to create, save, modify, and print various types of graphics.

Using the keyboard or optional light pen, you can combine a virtually unlimited number of predefined and user-defined symbols, freehand objects, and text labels into a drawing. You can direct output to a graphics printer, and an optional driver is available for the HP 7470A and 7475A plotters. *Jr-Draw* seems most suited for technical drawings, layouts, or business-type graphics.

Assembling Symbols Into Drawings

You create drawings by typing two-keystroke combinations to select and modify primitive symbols, from which more complex shapes are assembled. For example, typing ALT-S followed by 10 places a circle (symbol number 10) in the drawing area of the screen. Once it's there, you can use the cursor control keys and function keys to move and change the size of the object. You can rotate objects in increments of 90 degrees—except for circles and ellipses. Another option is selective erasure.

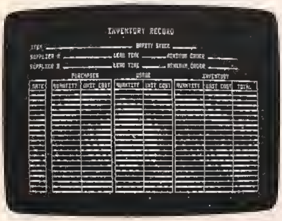


An office layout designed on a PCjr with Jr-Draw. *This sample screen is included with the software.*

Once created, adjacent objects can be selected together as if they were a single object, and all these manipulations can be performed on the group as a whole.

There are two ways to draw lines. The most flexible method is the freehand mode. You enter this mode by typing ALT-X, which converts the screen into something like an Etch-a-Sketch brand toy. As you move a crosshair around the screen with the cursor keys, a line is left in its wake.

I found myself using freehand mode almost exclusively. The second method requires you to press FN-4 at the beginning and end of each line segment to be plotted. Presumably this mode was intended for lines consisting of a single segment, but it's just as simple to use freehand mode for these as well.



INVENTORY RECORD

ITEM ________ SAFETY STOCK ____

SUPPLIER A ______ LEAD TIME ____ MINIMUM ORDER ____

SUPPLIER B ______ LEAD TIME ____ MINIMUM ORDER ____

	PURCHASES		USAGE		INVENTORY		
DATE	QUANTITY	UNIT COST	QUANTITY	UNIT COST	QUANTITY	UNIT COST	TOTAL

This inventory record chart is one of the predefined templates included on the Jr-Draw *disks.*

By combining these lines with the primitive symbols, pictures are built piece by piece. You can save the pictures on disk at any point.

Transferring To Paper

Ultimately, though, the object is to get these graphics onto paper. *Jr-Draw* offers eight different formats in which the drawing can be produced on any of a dozen graphics printers. Variations include the orientation of the drawing on the page and whether the drawing is printed in condensed, emphasized, or full-width typestyles.

Since a drawing can consist of up to 99 pages or screenfuls of information, you can also specify a range of pages to be printed at one time.

If you want a higher resolution copy, you can buy an optional driver for the plotters mentioned above. Using a plotter should minimize the jagged appearance of diagonal lines which characterizes graphics printed in screen resolution.

Jr-Draw comes with several symbol templates. They contain flow-charting symbols, electrical schematic symbols, large and small block text, and a few symbols designated "interior" for floor plans.

But the key to *Jr-Draw*'s flexibility lies in the ability to define custom symbol templates for specific applications. For instance, a template of architectural symbols might be useful for creating an elevation drawing. Or a band director might find a template of musical instruments helpful for charting seating arrangements.

Custom templates are created in much the same way as drawings—they're composed of previously defined symbols and freehand lines. Once the new combination is "compressed" and placed into the template, it can be used in defining yet another new symbol. Like drawings, these templates may be stored on disk.

A Little Confusion

Jr-Draw is a complex piece of software; it's not something which can be used intuitively. Fortunately, an extensive interactive tutorial spares you from having to read the entire 174-page reference manual before you start. The tutorial covers the program's basic operations.

Unfortunately, not everything in the tutorial works correctly. Furthermore, the manual states that the tutorial is on disk 2 (of the three disks provided with the package), when it's actually on disk 3. But overall, the tutorial is a useful feature and can be covered completely in a little over two hours.

Once beyond the tutorial, you'll find that unless you use *Jr-Draw* regularly and frequently, the quick reference card will be a necessity. It is expecting a lot of a user, for example, to remember that small block text should be spaced six units apart while large text is spaced 32 units apart. If any program ever begged for a keyboard overlay, *Jr-Draw* is it. On the plus side, *Jr-Draw* wisely displays the meanings of the ten function keys along the bottom of the screen.

Jr-Draw never crashed during testing, but there were several instances—although minor and correctable—when results did not match what the manual indicates should happen. For example, changing the aspect of an ellipse so that it was flattened horizontally resulted in it springing to a vertical orientation. And the TAB and ENTER keys did not work as described when adding text to a drawing.

Inadvertent keystrokes can also cause problems. Typing the BACKSPACE key caused the template to disappear, for example. It took several moments scanning through the manual to learn that the way to restore it was to type CTRL-H.

Sometimes the corrective action itself is a source of aggravation. If you try to fill with color an object that is not completely enclosed, it "springs a leak" and the entire screen is filled. The only remedy is to delete the object, redraw the screen, and recreate the object.

Would A Mac Be Better?

User feedback is, in general, good. Typically, the object or objects selected for manipulation blink on and off to distinguish them from other objects in the drawing. As these objects become numerous or complex, however, the blinking slows down. Eventually, you reach the point where there is a significant lag between a keystroke and a screen update. In most instances, though, this is not a serious problem.

There were moments, brief but real, when I wondered if a Macintosh with *MacPaint* would be better for the job. The Macintosh mouse and pull-down menus make it very easy to manipulate. Presumably, *Jr-Draw* would be much easier to use with the optional light pen instead of the keyboard, but I lacked a light pen for testing.

Only one other annoyance was encountered: *Jr-Draw* requires you to frequently interchange the program and data disks when moving from one menu to another. *Jr-Draw* is a good candidate for conversion to cartridge, which would eliminate this drawback.

The disks are not copy-protected, but neither the manual nor the tutorial emphasizes the importance of backing up the disks before proceeding (this information is in Appendix B of the manual—read it *first*). The manual recommends everyday use of the original disk and setting aside the copies for backups, just the opposite of what most experts advise. Make sure your backups really work before following this practice.

Practical Applications

It is reasonable to use a computer to create drawings only when the computer offers some advantages over conventional methods. It may be that drawings can be created more quickly on a computer, or that once created, they are more easily modified. Or perhaps the quality of the drawings is improved, or the drawings can be produced more cost-effectively.

The answers to these issues depend partially on the specific software, but to a larger degree on the environment in which the software will be operated.

A site with no flat-art capability yet a need for casual graphics such as organizational charts may find *Jr-Draw* a useful tool. A one-page chart can be created in less than half an hour, and changes or updates are easily made.

But it should be understood that *Jr-Draw* produces graphics suitable for use in reports to other members of your department, perhaps, but not necessarily for sale to clients or for presentation to a board of directors.

There are many graphics programs on the market for the PC and PCjr. One of the worthy competitors to *Jr-Draw* is IBM's own *ColorPaint* program. PCjr owners should consider several different systems before selecting one to meet their needs.

Jr-Draw
Micrografx
1701 N. Greenville Avenue
Suite 703
Richardson, TX 75081
$195

©

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.	1.	*Flight Simulator II*	SubLogic	Aircraft simulation	●	●	●		
2.		*Karateka*	Brøderbund	Action karate game	●		●		
3.	4.	*F-15 Strike Eagle*	MicroProse	Air combat simulation	●	●	●	●	
4.	2.	*The Hitchhiker's Guide To The Galaxy*	Infocom	Comic adventure strategy game	●	●	●	●	●
5.	3.	*Flight Simulator*	Microsoft	Aircraft simulation				●	
Education									
1.	3.	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	●	●	●	●	
2.	2.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	●		●	●	●
3.	1.	*New Improved MasterType*	Scarborough	Typing instruction program	●	●	●	●	●
4.	4.	*Early Games*	Springboard	Educational games, ages 2-6	●	●	●	●	
5.		*Bank Street Music Writer*	Mindscape	Music composition program	●	●	●		
Home Management									
1.	1.	*Print Shop*	Brøderbund	Do-it-yourself print shop	●	●	●		
2.	2.	*Print Shop Graphics Library*	Brøderbund	100 additional graphics	●	●	●		
3.	4.	*HomePak*	Batteries Included	Word processing, telecommunications, & data management	●	●	●	●	
4.	5.	*Andrew Tobias's Managing Your Money*	Meca	Home financial program				●	
5.		*The Newsroom*	Springboard	Do-it-yourself newspaper	●		●	●	

Copyright 1985 by Billboard Publications, Inc. Compiled by the Billboard Research Department and reprinted by permission. Data as of 7/06/85.

Commodore 64 Memory Manager

Robert Lee

If you find yourself using several BASIC programs repeatedly, here's a way you can load them all into your computer at once, and run them independently. "Memory Manager" keeps track of up to eight programs in your Commodore 64 and lets you switch between them with the special function keys.

The Commodore 64 has 38K of Random Access Memory (RAM) available for BASIC programs. However, unless you're using a very large program, most of that memory is sitting empty, wasted.

"Memory Manager" is a utility which takes advantage of the leftover memory by using it to store other BASIC programs. It also uses 8K of additional RAM which is hidden beneath the Read Only Memory (ROM). Normally, this ROM prevents you from using the additional RAM, but Memory Manager collects every available byte of RAM (49.5K total) and partitions it into eight sections. You can load, list, run, and save up to eight BASIC programs in your computer with Memory Manager.

To use Memory Manager, type in and run the accompanying program. It asks you for the maximum amount of memory (in kilobytes) to be reserved for BASIC. The default response printed on the screen for you is 9K; simply press RETURN, or enter another value if you like. You can't change this value later without restarting the computer, so your response defines the maximum size of the BASIC program you can run. If you aren't sure how long your programs are, you can make a close estimate if you have a disk drive. Load a disk directory and note the number of blocks the program consumes on the disk. Since each block equals 256 bytes, four blocks equal one kilobyte. Simply divide the number of blocks by four to estimate the length. (For instance, a program that is 25 blocks long on the directory takes about 6.25K of RAM.) However, keep in mind that some programs require additional RAM when they run.

After you enter your answer, the cursor reappears and Memory Manager is ready to run. Activate it by typing SYS 53128 and pressing RETURN.

Eight Partitions

Depending on the amount of memory space available, up to eight programs can be handled by Memory Manager. The partitions are accessed by pressing one of the four special function keys. Press f1 to access partition 1, f2 for partition 2, and so on. When you flip to a different partition, Memory Manager displays the partition number on the screen.

For example, try typing or loading a program into the computer. This is partition 1. Type LIST to confirm that it's in memory. Now press one of the function keys—say, f5. When you type LIST again, nothing's there. To fill partition 5, just type or load another program. You can switch from partition to partition as often as you like. (If you press f5 when you're already in partition 5, nothing happens.)

Memory Manager uses only the space required to store a program, so none is wasted. If there is not enough room to store a certain program, Memory Manager delivers an error message.

If you wish to deactivate Memory Manager for some reason, type SYS 53144 and press RETURN. Pressing the RUN/ STOP–RESTORE combination also disables Memory Manager. You can turn it on again by entering SYS 53128. All the programs in memory will remain intact—although they may be damaged if you perform other tasks while Memory Manager is deactivated.

Remember that Memory Manager works only with BASIC programs; machine language programs are almost sure to cause memory conflicts. (The machine language portion of Memory Manager is stored above address 52736, $CE00 hex. It frees up RAM from $0800 to $CDFF minus the memory space assigned to BASIC.) Even with BASIC,

Sale

Commodore 64 Accessories

SAVE

☆ ☆ SUPER ☆ ☆

Summer Sizzler Sale

Sale Positively Ends 9-15-85

Prices will go back to regular sale prices	List	Reg. Sale Price	Summer Sizzler Sale
C128 Commodore Computer Expandable to 512K, runs C-64, CPM, and 7.0 Programs. (Add $10 Shipping)	$349.00	$299.09	$289.95*
15½" Commodore 150-170 CPS Printer Near Letter Quality, Multiple Pin Tractor / Friction Feed. Best Printer Value in U.S.A. (Add $17.50 Shipping)	$895.00	$299.00	$199.95*
Commodore-64 IEEE Interface Allows you to run Pet Peripherals on the C-64, including the One Megabyte Disk Drive and 15½" Printer	$109.95	$69.00	$65.95
Juki Printer/Typewriter Letter Quality, daisy wheel, use as typewriter and/or printer (auto correction) (Add $10 Shipping)	$349.00	$249.00	$229.95*
SCM 80 CPS Printer Tractor/Friction 10" Famous Name Printer does Graphics w/Interface. (Add $10 Shipping)	$299.00	$159.00	$149.95*
Cardco G Plus Interface Converts Commodore to centronics for use with most printers, plus does Commodore graphics on graphic printers	$109.00	$59.00	$49.95
Alphacom 40 Column Printer Thermol technology - does graphics. (Add $7.50 Shipping) *Alphacom C-64 or Atari Interface $8.95.*	$99.00	$24.95	$22.95
190K Slim-Line Disk Drive Cooler, 20% faster, quieter than 1541 drive (Add $10 Shipping)	$249.00	$169.00	$139.95*
One Megabyte Disk Drive (1000K) Double sided drive hooks up to C-64 with IEEE interface, perfect as a second drive. (Add $10 Shipping)	$889.00	$199.00	$179.95*
13" Premium Quality Color Monitor Better than 1702, with separated video inputs. (Add $14.50 Shipping)	$399.00	$199.00	$169.95*
Voice Synthesizer Allows you to talk through your computer. Optional software lets you play talking adventure games (Zork, etc.)	$89.00	$49.00	$39.95
80 Column Board - 4 Slot Expander Allows you to program in 80 columns, plus gives you a 4 slot switch selectable expander	$149.00	$79.00	$59.95
Auto Dial Modem with Super Smart Software Upload, Download, Print, Catalog, This Package has it all!!! Best Modem Value in the U.S.A.!!	$129.95	$59.00	$39.95
Musicalc I Software The Best Musical Software for the C-64, allows you to change all parameters.	$59.95	$29.95	$14.95
Oil Barrons Software Better than Monopoly, comes with game board, disks and instruction manual. Strike Oil or Live in the Poor Farm	$49.95	$19.95	$9.95

*** plus Software Coupon Discounts**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order

keeping the programs from interfering with each other in every instance is practically impossible. BASIC programs with machine language subroutines, custom character sets, or POKEs into memory locations beyond the top of BASIC memory can mess up the programs stored in other partitions.

Variables set to certain values by a program in one partition will retain those values when you switch to another partition (although they'll be reset when you type RUN). For these reasons, we don't recommend using Memory Manager for critical applications such as software development. Instead, it's more suitable for keeping frequently used programs in memory rather than constantly accessing the cassette or disk drive, or for loading up a series of programs for a young person who cannot handle tapes or disks.

Commodore 64 Memory Manager

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
10 PRINT"{CLR}{6 DOWN}
   {11 RIGHT}MEMORY MANAGER"
                          :rem 62
20 PRINT"{3 DOWN}{11 RIGHT}
   {3 SPACES}FOR THE ":rem 109
30 PRINT"{3 DOWN}{12 RIGHT}COM
   MODORE 64{2 SPACES}"
                         :rem 210
100 FORX=52736TO53215 :rem 183
110 READA:CK=CK+A:POKEX,A
                          :rem 28
120 NEXT                 :rem 210
130 IF CK<>68936 THEN PRINT"
    {RVS}{2 DOWN} ERROR IN DAT
    A STATEMENTS":STOP :rem 50
140 INPUT"{5 DOWN}HOW MANY K F
    OR PROGRAM (6 TO 24)
    {2 RIGHT}9{3 LEFT}";M
                         :rem 141
145 IFM<6ORM>25THENPRINT"{CLR}
    NUMBER SHOULD BE FROM 6 TO
     24":GOTO140         :rem 168
150 POKE55,0:POKE56,M*4+8
                         :rem 153
160 FORX=53224TO53231:POKEX,M*
    4+8:POKEX+16,M*4+8:NEXT
                         :rem 181
170 FORX=0TO6:POKEX+53217,X*3+
    1:POKEX+53233,X*3+4:NEXT
                         :rem 237
180 POKE53214,X*3+1:POKE53215,
    M*4+8                :rem 167
190 FORX=(M*4+8)*256+1TO(M*4+8
    )*256+24:POKEX,0:NEXT
                         :rem 136
200 PRINT"{CLR}{5 DOWN}
    {7 RIGHT}SYS 53128 TO ACTI
    VATE"                 :rem 12
210 PRINT"{3 DOWN}{7 RIGHT}SYS
     53144 TO DEACTIVATE"
                         :rem 223
220 PRINT"{3 DOWN}{9 RIGHT}PRO
    GRAM #1 IN USE"   :rem 141
230 PRINT"{4 DOWN}SYS 53128
    {3 UP}"               :rem 95
52736 DATA 169,255,141,180,207
      ,162               :rem 154
52742 DATA 19,189,181,207,32,2
      10                  :rem 49
52748 DATA 255,202,16,247,88,7
      6                   :rem 17
52754 DATA 49,234,162,255,165,
      157                :rem 113
52760 DATA 240,247,165,203,201
      ,64                 :rem 91
52766 DATA 208,5,141,180,207,2
      40                  :rem 45
52772 DATA 236,172,180,207,192
      ,64                :rem 106
52778 DATA 208,229,201,3,208,2
                         :rem 204
52784 DATA 162,6,201,4,208,2
                          :rem 98
52790 DATA 162,0,201,5,208,2
                          :rem 90
52796 DATA 162,2,201,6,208,2
                          :rem 99
52802 DATA 162,4,224,255,240,2
      01                  :rem 33
52808 DATA 173,141,2,240,1,232
                         :rem 190
52814 DATA 236,221,207,240,190
      ,120               :rem 134
52820 DATA 160,8,132,88,160,0
                         :rem 147
52826 DATA 132,87,173,222,207,
      133                 :rem 99
52832 DATA 89,173,223,207,133,
      90                  :rem 54
52838 DATA 134,91,162,3,165,90
                         :rem 211
52844 DATA 201,206,240,144,177
      ,87                :rem 101
52850 DATA 145,89,230,87,208,2
                         :rem 213
52856 DATA 230,88,230,89,208,2
                         :rem 215
52862 DATA 230,90,201,0,208,22
      8                  :rem 244
52868 DATA 202,208,227,165,1,4
      1                  :rem 254
52874 DATA 254,133,1,166,91,18
      9                   :rem 12
52880 DATA 240,207,56,253,224,
      207                 :rem 98
52886 DATA 133,87,189,248,207,
      253                :rem 124
52892 DATA 232,207,133,88,172,
      221                :rem 102
52898 DATA 207,173,222,207,153
      ,224               :rem 154
52904 DATA 207,173,223,207,153
      ,232               :rem 142
52910 DATA 207,165,89,153,240,
      207                :rem 101
52916 DATA 165,90,153,248,207,
      160                :rem 105
52922 DATA 7,185,232,207,221,2
      48                  :rem 50
52928 DATA 207,144,44,208,8,18
      5                   :rem 10
52934 DATA 224,207,221,240,207
      ,144               :rem 139
52940 DATA 34,185,224,207,56,2
      29                  :rem 56
52946 DATA 87,153,224,207,185,
      232                :rem 110
52952 DATA 207,229,88,153,232,
      207                :rem 108
52958 DATA 185,240,207,56,229,
      87                  :rem 71
52964 DATA 153,240,207,185,248
      ,207               :rem 157
52970 DATA 229,88,153,248,207,
      136                :rem 116
52976 DATA 16,201,189,224,207,
      133                :rem 103
52982 DATA 94,189,232,207,133,
      95                  :rem 68
52988 DATA 169,0,133,87,169,8
                         :rem 180
52994 DATA 133,88,189,240,207,
      133                :rem 114
53000 DATA 92,189,248,207,133,
      93                  :rem 53
53006 DATA 160,0,177,94,145,87
                         :rem 208
53012 DATA 230,87,208,2,230,88
                         :rem 198
53018 DATA 230,94,208,2,230,95
                         :rem 200
53024 DATA 165,95,197,93,208,2
      34                  :rem 62
53030 DATA 165,94,197,92,208,2
      28                  :rem 60
53036 DATA 189,224,207,133,87,
      189                :rem 114
53042 DATA 232,207,133,88,177,
      94                  :rem 55
53048 DATA 145,87,230,87,208,2
                         :rem 211
53054 DATA 230,88,230,94,208,2
                         :rem 202
53060 DATA 230,95,165,95,197,9
      0                   :rem 11
53066 DATA 208,234,165,94,197,
      89                  :rem 72
53072 DATA 208,228,172,221,207
      ,185               :rem 147
53078 DATA 240,207,141,222,207
      ,185               :rem 143
53084 DATA 248,207,141,223,207
      ,142               :rem 142
53090 DATA 221,207,165,1,9,1
                          :rem 92
53096 DATA 133,1,173,221,207,2
      4                  :rem 244
53102 DATA 105,49,141,209,207,
      162                 :rem 87
53108 DATA 19,189,201,207,32,2
      10                  :rem 39
53114 DATA 255,202,16,247,169,
      255                :rem 100
53120 DATA 141,180,207,88,76,4
      9                    :rem 3
53126 DATA 234,0,120,169,20,14
      1                  :rem 235
53132 DATA 20,3,169,206,141,21
                         :rem 186
53138 DATA 3,88,96,0,0,0
                         :rem 156
53144 DATA 120,169,49,141,20,3
                         :rem 194
53150 DATA 169,234,141,21,3,88
                         :rem 201
53156 DATA 96,0,0,0,0,255
                         :rem 197
53162 DATA 0,255,0,255,0,255
                          :rem 91
53168 DATA 0,255,0,255,64,141
                         :rem 149
53174 DATA 89,82,79,77,69,77
                         :rem 144
53180 DATA 32,72,71,85,79,78
                         :rem 121
53186 DATA 69,32,84,79,78,141
                         :rem 178
53192 DATA 147,141,69,83,85,32
                         :rem 216
53198 DATA 78,73,32,49,35,32
                         :rem 120
53204 DATA 77,65,82,71,79,82
                         :rem 121
53210 DATA 80,141,147,0,21,204
                         :rem 180
```

©

ATARI®

88K *Lowest Price In The USA!* 152K

Computer System Sale

• Students • Word Processing • Home • Business

[1] *Rated "Best Buy" by Consumers Digest Buyers Guide, January 1985*

LOOK AT ALL YOU GET FOR ONLY **$449** SYSTEM PRICE

	LIST PRICE	INDIVIDUAL SALE PRICE
Atari 800XL 88K Computer	$179.00	$109.00
Atari 1050 127K Disk Drive	299.00	199.00
Atari 1027 Letter Quality 20 CPS Printer	299.00	199.00
Atari Writer Word Processer	59.95	39.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	$852.90	$559.90

All connecting cables & T.V. interface included.
Monitors sold separetly.

SAVE $100
All 5 ONLY
$449.00
SYSTEM SALE PRICE

152K SYSTEM
$499.00

Other Accessories

	List	Sale
12" Hi Resolution Green or Amber Screen Monitor	$199.00	69.95
13" Hi Resolution Color Monitor	$399.00	169.95
* ATARI 130XE 152K Computer	$249.00	159.00

Add $9.95 for Connection Cables (Monitors Only)
(Ltd. Qty)
Add $10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY.** If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D.

COMPUTER DIRECT
We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S̲. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S̲}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <	↖	Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◂	Backspace
{DELETE}	ESC CTRL DELETE	◧	Delete character
{INSERT}	ESC CTRL INSERT	◨	Insert character
{DEL LINE}	ESC SHIFT DELETE	⊟	Delete line
{INS LINE}	ESC SHIFT INSERT	⊞	Insert line
{TAB}	ESC TAB	▸	TAB key
{CLR TAB}	ESC CTRL TAB	◧	Clear tab
{SET TAB}	ESC SHIFT TAB	◨	Set tab stop
{BELL}	ESC CTRL 2	◳	Ring buzzer
{ESC}	ESC ESC	⭠	ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME	♥	[<1>]	COMMODORE 1	✚
{HOME}	CLR/HOME	S	[<2>]	COMMODORE 2	◢
{UP}	SHIFT ↑ CRSR ↓	●	[<3>]	COMMODORE 3	▩
{DOWN}	↑ CRSR ↓	Q	[<4>]	COMMODORE 4	▣
{LEFT}	SHIFT ← CRSR →	▌	[<5>]	COMMODORE 5	⊥
{RIGHT}	← CRSR →	]	[<6>]	COMMODORE 6	▐
{RVS}	CTRL 9	R	[<7>]	COMMODORE 7	○
{OFF}	CTRL 0	▬	[<8>]	COMMODORE 8	▚
{BLK}	CTRL 1	▒	{ F1 }	f1	▬
{WHT}	CTRL 2	E	{ F2 }	SHIFT f1	╮
{RED}	CTRL 3	£	{ F3 }	f3	▬
{CYN}	CTRL 4	◥	{ F4 }	SHIFT f3	╰
{PUR}	CTRL 5	▒	{ F5 }	f5	▌
{GRN}	CTRL 6	↑	{ F6 }	SHIFT f5	╯
{BLU}	CTRL 7	←	{ F7 }	f7	▐
{YEL}	CTRL 8	π	{ F8 }	SHIFT f7	▄
			←	←	▒

The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program. The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a decimal number (on the Commodore), a hexadecimal number (on the Apple), or a pair of letters (on the Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, Atari, and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Commodore and Atari Proofreaders do not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it *will* detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you type NEW, the Proofreader prompts you to press Y to be sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program in BASIC as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. The version of your program that you resave from BASIC will take up less space on disk and will load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP–RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10
 SPACES}":FOR X=1 TO 4:A$=A$
 +B$:NEXT
FOR X=886 TO 1018:A$=A$+CHR$
(PEEK(X)):NEXT:OPEN 1,1,1,A$:
 CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a paritally completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY. You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"{CLR}PLEASE WAIT...":
   FORI=886TO1018:READA:CK=CK+
   A:POKEI,A:NEXT
20 IF CK<>17539 THEN PRINT"
   {DOWN}YOU MADE AN ERROR":PR
   INT"IN DATA STATEMENTS.":EN
   D
30 SYS886:PRINT"{CLR}{2 DOWN}P
   ROOFREADER ACTIVATED.":NEW
40 DATA 173,036,003,201,150,20
   8,001,096,141,151,003,173
50 DATA 037,003,141,152,003,16
   9,150,141,036,003,169,003
60 DATA 141,037,003,169,000,13
   3,254,096,032,087,241,133
70 DATA 251,134,252,132,253,00
   8,201,013,240,017,201,032
80 DATA 240,005,024,101,254,13
   3,254,165,251,166,252,164
90 DATA 253,040,096,169,013,03
   2,210,255,165,214,141,251
100 DATA 003,206,251,003,169,0
    00,133,216,169,019,032,210
110 DATA 255,169,018,032,210,2
    55,169,58,032,210,255,166
120 DATA 254,169,000,133,254,1
    72,151,003,192,087,208,006
130 DATA 032,205,189,076,235,0
    03,032,205,221,169,032,032
140 DATA 210,255,032,210,255,1
    73,251,003,133,214,076,173
150 DATA 003
```

Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
120 IF CK<>19072 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
130 A=USR(1536)
140 ? :? "Automatic Proof
    reader Now Activated.
    "
150 END
160 DATA 104,160,0,185,26
    ,3,201,69,240,7
170 DATA 200,200,192,34,2
    08,243,96,200,169,74
180 DATA 153,26,3,200,169
    ,6,153,26,3,162
190 DATA 0,189,0,228,157,
    74,6,232,224,16
200 DATA 208,245,169,93,1
    41,78,6,169,6,141
210 DATA 79,6,24,173,4,22
    8,105,1,141,95
```

```
220 DATA 6,173,5,228,105,
    Ø,141,96,6,169
230 DATA Ø,133,2Ø3,96,247
    ,238,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,2Ø5,23B
250 DATA Ø,Ø,Ø,Ø,Ø,32,62,
    246,B,2Ø1
260 OATA 155,24Ø,13,2Ø1,3
    2,24Ø,7,72,24,1Ø1
270 DATA 2Ø3,133,2Ø3,1Ø4,
    4Ø,96,72,152,72,138
280 DATA 72,16Ø,Ø,169,128
    ,145,88,2ØØ,192,4Ø
290 DATA 2Ø8,249,165,2Ø3,
    74,74,74,74,24,1Ø5
300 DATA 161,16Ø,3,145,88
    ,165,2Ø3,41,15,24
310 DATA 1Ø5,161,2ØØ,145,
    88,169,Ø,133,2Ø3,1Ø4
320 OATA 17Ø,1Ø4,168,1Ø4,
    4Ø,96
```

Program 3: IBM Proofreader

By Charles Brannon, Program Editor

```
1Ø 'Automatic Proofreader Ver
   sion 2.ØØ (Lines 27Ø,51Ø,5
   15,517,62Ø,63Ø changed fro
   m V1.Ø)
1ØØ OIM L$(5ØØ),LNUM(5ØØ):COL
    OR Ø,7,7:KEY OFF:CLS:MAX=
    Ø:LNUM(Ø)=65536!
11Ø ON ERROR GOTO 12Ø:KEY 15,
    CHR$(4)+CHR$(7Ø):ON KEY(1
    5) GOSUB 64Ø:KEY (15) ON:
    GOTO 13Ø
12Ø RESUME 13Ø
13Ø OEF SEG=&H4Ø:W=PEEK(&H4A)
14Ø ON ERROR GOTO 65Ø:PRINT:P
    RINT"Proofreader Ready."
15Ø LINE INPUT L$:Y=CSRLIN-IN
    T(LEN(L$)/W)-1:LOCATE Y,1
16Ø DEF SEG=Ø:POKE 1Ø5Ø,3Ø:PO
    KE 1Ø52,34:POKE 1Ø54,Ø:PO
    KE 1Ø55,79:POKE 1Ø56,13:P
    OKE 1Ø57,2B:LINE INPUT L$
    :DEF SEG:IF L$="" THEN 15
    Ø
17Ø IF LEFT$(L$,1)=" " THEN L
    $=MID$(L$,2):GOTO 17Ø
18Ø IF VAL(LEFT$(L$,2))=Ø AND
     MID$(L$,3,1)=" " THEN L$
    =MID$(L$,4)
19Ø LNUM=VAL(L$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
2ØØ IF ASC(L$)>57 THEN 26Ø 'n
    o line number, therefore
    command
21Ø IF TEXT$="" THEN GOSUB 54
    Ø:IF LNUM=LNUM(P) THEN GO
    SUB 56Ø:GOTO 15Ø ELSE 15Ø
22Ø CKSUM=Ø:FOR I=1 TO LEN(L$
    ):CKSUM=(CKSUM+ASC(MID$(L
    $,I))*I) ANO 255:NEXT:LOC
    ATE Y,1:PRINT CHR$(65+CKS
    UM/16)+CHR$(65+(CKSUM AND
     15))+" "+L$
23Ø GOSUB 54Ø:IF LNUM(P)=LNUM
     THEN L$(P)=TEXT$:GOTO 15
    Ø 'replace line
24Ø GOSUB 58Ø:GOTO 15Ø 'inser
    t the line
26Ø TEXT$="":FOR I=1 TO LEN(L
    $):A=ASC(MID$(L$,I)):TEXT
    $=TEXT$+CHR$(A+32*(A>96 A
    ND A<123)):NEXT
27Ø OELIMITER=INSTR(TEXT$," "
    ):COMMANO$=TEXT$:ARG$="":
    IF OELIMITER THEN COMMANO
    $=LEFT$(TEXT$,OELIMITER-1
    ):ARG$=MIO$(TEXT$,OELIMIT
    ER+1) ELSE OELIMITER=INST
    R(TEXT$,CHR$(34)):IF OELI
    MITER THEN COMMAND$=LEFT$
    (TEXT$,OELIMITER-1):ARG$=
    MIO$(TEXT$,OELIMITER)
28Ø IF COMMANO$<>"LIST" THEN
    41Ø
29Ø OPEN "scrn:" FOR OUTPUT A
    S #1
3ØØ IF ARG$="" THEN FIRST=Ø:P
    =MAX-1:GOTO 34Ø
31Ø DELIMITER=INSTR(ARG$,"-")
    :IF OELIMITER=Ø THEN LNUM
    =VAL(ARG$):GOSU8 54Ø:FIRS
    T=P:GOTO 34Ø
32Ø FIRST=VAL(LEFT$(ARG$,OELI
    MITER)):LAST=VAL(MIO$(ARG
    $,OELIMITER+1))
33Ø LNUM=FIRST:GOSU8 54Ø:FIRS
    T=P:LNUM=LAST:GOSU8 54Ø:I
    F P=Ø THEN P=MAX-1
34Ø FOR X=FIRST TO P:N$=MIO$(
    STR$(LNUM(X)),2)+" "
35Ø IF CKFLAG=Ø THEN A$="":GO
    TO 37Ø
36Ø CKSUM=Ø:A$=N$+L$(X):FOR I
    =1 TO LEN(A$):CKSUM=(CKSU
    M+ASC(MIO$(A$,I))*I) ANO
    255:NEXT:A$=CHR$(65+CKSUM
    /16)+CHR$(65+(CKSUM ANO 1
    5))+" "
37Ø PRINT #1,A$+N$+L$(X)
38Ø IF INKEY$<>"" THEN X=P
39Ø NEXT :CLOSE #1:CKFLAG=Ø
4ØØ GOTO 13Ø
41Ø IF COMMAND$="LLIST" THEN
    OPEN "Ipt1:" FOR OUTPUT A
    S #1:GOTO 3ØØ
42Ø IF COMMAND$="CHECK" THEN
    CKFLAG=1:GOTO 29Ø
43Ø IF COMMAND$<>"SAVE" THEN
    45Ø
44Ø GOSU8 6ØØ:OPEN ARG$ FOR O
    UTPUT AS #1:ARG$="":GOTO
    3ØØ
45Ø IF COMMAND$<>"LOAD" THEN
    49Ø
46Ø GOSUB 6ØØ:OPEN ARG$ FOR I
    NPUT AS #1:MAX=Ø:P=Ø
47Ø WHILE NOT EOF(1):LINE INP
    UT #1,L$:LNUM(P)=VAL(L$):
    L$(P)=MID$(L$,LEN(STR$(VA
    L(L$)))+1):P=P+1:WEND
48Ø MAX=P:CLOSE #1:GOTO 13Ø
49Ø IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure";L$:IF LEFT$(L$,
    1)="y" OR LEFT$(L$,1)="Y"
     THEN MAX=Ø:GOTO 13Ø:ELSE
     13Ø
5ØØ IF COMMAND$="BASIC" THEN
    COLOR 7,Ø,Ø:ON ERROR GOTO
     Ø:CLS:END
51Ø IF COMMAND$<>"FILES" THEN
     52Ø
515 IF ARG$="" THEN ARG$="A:"
     ELSE SEL=1:GOSUB 6ØØ
517 FILES ARG$:GOTO 13Ø
52Ø PRINT"Syntax error":GOTO
    13Ø
54Ø P=Ø:WHILE LNUM>LNUM(P) AN
    D P<MAX:P=P+1:WENO:RETURN
56Ø MAX=MAX-1:FOR X=P TO MAX:
    LNUM(X)=LNUM(X+1):L$(X)=L
    $(X+1):NEXT:RETURN
5B0 MAX=MAX+1:FOR X=MAX TO P+
    1 STEP -1:LNUM(X)=LNUM(X-
    1):L$(X)=L$(X-1):NEXT:L$(
    P)=TEXT$:LNUM(P)=LNUM:RET
    URN
6ØØ IF LEFT$(ARG$,1)<>CHR$(34
    ) THEN 52Ø ELSE ARG$=MIO$
    (ARG$,2)
61Ø IF RIGHT$(ARG$,1)=CHR$(34
    ) THEN ARG$=LEFT$(ARG$,LE
    N(ARG$)-1)
62Ø IF SEL=Ø AND INSTR(ARG$,"
    .")=Ø THEN ARG$=ARG$+".8A
    S"
63Ø SEL=Ø:RETURN
64Ø CLOSE #1:CKFLAG=Ø:PRINT"S
    topped.":RETURN 15Ø
65Ø PRINT "Error #";ERR:RESUM
    E 15Ø
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
1Ø C = Ø: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
2Ø IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER OATA STAT
   EMENTS": END
3Ø IF PEEK (19Ø * 256) < > 76 T
   HEN POKE 56,Ø: POKE 57,3: CA
   LL 1ØØ2: GOTO 5Ø
4Ø PRINT CHR$ (4);"IN#A$3ØØ"
5Ø POKE 34,Ø: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREAOER
   INSTALLED"
6Ø NEW
1ØØ DATA 216,32,27,253,2Ø1,141
11Ø DATA 2ØB,6Ø,13B,72,169,Ø
12Ø DATA 72,1B9,255,1,2Ø1,16Ø
13Ø DATA 24Ø,B,1Ø4,1Ø,125,255
14Ø OATA 1,1Ø5,Ø,72,2Ø2,2ØB
15Ø DATA 23B,1Ø4,17Ø,41,15,9
16Ø OATA 48,2Ø1,5B,144,2,233
17Ø DATA 57,141,1,4,13B,74
1BØ DATA 74,74,74,41,15,9
19Ø DATA 4B,2Ø1,5B,144,2,233
2ØØ OATA 57,141,Ø,4,1Ø4,17Ø
21Ø DATA 169,141,96
```

©

MLX

Machine Language Entry Program For Commodore 64 and Atari

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. Both versions of MLX asks you for two numbers: the starting address and the ending address. In addition, the Atari version asks for a run/init address. These numbers are given in the article accompanying the ML program presented in MLX format. The Atari version also gives you three options for saving the file: as a boot tape, as disk binary file, or as boot disk. The article with the ML program should suggest which format to use.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the Commodore INST/DEL key or the Atari DEL/BACK SPACE; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, the Commodore 64 version of MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

	U	I	O			7	8	9
H	J	K	L	become	0	4	5	6
	M	,	.			1	2	3

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. Each command is accessed by pressing one letter, plus the SHIFT key for 64 MLX or the CTRL key for the Atari version. MLX recognizes these commands:

Commodore	Atari	Command
SHIFT-S	CTRL-S	Save
SHIFT-L	CTRL-L	Load
SHIFT-N	CTRL-N	New Address
SHIFT-D	CTRL-D	Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember to make a note of what address you stop at. The next time you run MLX, answer all the prompts as you did before—regardless of where you stopped typing—then insert the disk or tape. When you get to the entry prompt, press SHIFT-L (64) or CTRL-L (Atari) to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N (64) or CTRL-N (Atari) and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the MLX-format listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D or CTRL-D, enter two addresses within the line number range of the listing. You can break out of the listing display and return to the prompt by pressing any key.

Atari MLX: Machine Language Entry

```
DA 100 GRAPHICS 0:DL=PEEK(56
       0)+256*PEEK(561)+4:PD
       KE DL-1,71:PDKE DL+2,
       6
NJ 110 PDSITIDN B,0:? "MLX":
       PDSITIDN 23,0:? "fail
       safe entry":PDKE 710,
       0:?
JK 120 ? "Starting Address";
       :INPUT BEG:? "  Endin
       g Address";:INPUT FIN
       :? "Run/Init Address"
       ;:INPUT STARTADR
DD 130 DIM A(6),BUFFER$(FIN-
       BEG+127),T$(20),F$(20
       ),CID$(7),SECTDR$(12B
       ),DSKINV$(6)
JJ 140 DPEN #1,4,0,"K:":? :?
        ,"Tape or Disk:";
BM 150 BUFFER$=CHR$(0):BUFFE
       R$(FIN-BEG+30)=BUFFER
       $:BUFFER$(2)=BUFFER$:
       SECTDR$=BUFFER$
GC 160 ADDR=BEG:CID$="hhh":C
       ID$(4)=CHR$(170):CID$
       (5)="LV":CID$(7)=CHR$
       (22B)
EJ 170 GET #1,MEDIA:IF MEDIA
       <>B4 AND MEDIA<>6B TH
       EN 170
PO 1B0 ? CHR$(MEDIA):? :IF M
       EDIA<>ASC("T") THEN B
       UFFER$="":GDTD 250
PL 190 BEG=BEG-24:BUFFER$=CH
       R$(0):BUFFER$(2)=CHR$
       (INT((FIN-BEG+127)/12
       B))
KF 200 H=INT(BEG/256):L=BEG-
       H*256:BUFFER$(3)=CHR$
       (L):BUFFER$(4)=CHR$(H
       )
EC 210 PINIT=BEG+B:H=INT(PIN
       IT/256):L=PINIT-H*256
       :BUFFER$(5)=CHR$(L):B
       UFFER$(6)=CHR$(H)
PB 220 FDR I=7 TD 24:READ A:
       BUFFER$(I)=CHR$(A):NE
       XT I:DATA 24,96,169,6
       0,141,2,211,169,0,133
       ,10,169,0,133,11,76,0
       ,0
DP 230 H=INT(STARTADR/256):L
       =STARTADR-H*256:BUFFE
       R$(15)=CHR$(L):BUFFER
       $(19)=CHR$(H)
KL 240 BUFFER$(23)=CHR$(L):B
       UFFER$(24)=CHR$(H)
HI 250 IF MEDIA<>ASC("D") TH
       EN 360
OD 260 ? :? "Boot Disk or Bi
       nary File:";
LI 270 GET #1,DTYPE:1F DTYPE
       <>6B AND DTYPE<>70 TH
       EN 270
GM 2B0 ? CHR$(DTYPE):IF DTYP
       E=70 THEN 360
PJ 290 BEG=BEG-30:BUFFER$=CH
       R$(0):BUFFER$(2)=CHR$
       (INT((FIN-BEG+127)/12
       B))
KG 300 H=INT(BEG/256):L=BEG-
       H*256:BUFFER$(3)=CHR$
       (L):BUFFER$(4)=CHR$(H
       )
HH 310 PINIT=STARTADR:H=INT(
       PINIT/256):L=PINIT-H*
       256:BUFFER$(5)=CHR$(L
       ):BUFFER$(6)=CHR$(H)
AO 320 RESTDRE 330:FDR 1=7 T
       D 30:READ A:BUFFER$(I
       )=CHR$(A):NEXT I
GA 330 DATA 169,0,141,231,2,
       133,14,169,0,141,232,
       2,133,15,169,0,133,10
       ,169,0,133,11,24,96
OB 340 H=INT(BEG/256):L=BEG-
       H*256:BUFFER$(B)=CHR$
       (L):BUFFER$(15)=CHR$(
       H)
DD 350 H=INT(STARTADR/256):L
       =STARTADR-H*256:BUFFE
       R$(22)=CHR$(L):BUFFER
```

```
         $(26)=CHR$(H)
JP 36Ø   GRAPHICS Ø:POKE 712,1
         Ø:POKE 71Ø,1Ø:POKE 7Ø
         9,2
JK 37Ø   ? ADDR;":";:FOR J=1 T
         O 6
HF 38Ø   GOSUB 57Ø:IF N=-1 THE
         N J=J-1:GOTO 38Ø
BF 39Ø   IF N=-19 THEN 72Ø
DI 4ØØ   IF N=-12 THEN LET REA
         D=1:GOTO 72Ø
AI 41Ø   TRAP 41Ø:IF N=-14 THE
         N ? :? "New Address";
         :INPUT ADDR:? :GOTO 3
         7Ø
JD 42Ø   TRAP 32767:IF N<>-4 T
         HEN 48Ø
AJ 43Ø   TRAP 43Ø:? :? "Displa
         y:From";:INPUT F:? ,"
         To";:INPUT T:TRAP 327
         67
ML 44Ø   IF F<BEG OR F>FIN OR
         T<BEG OR T>FIN OR T<F
          THEN ? CHR$(253);"At
          least ";BEG;", Not M
         ore Than ";FIN:GOTO 4
         3Ø
MM 45Ø   FOR I=F TO T STEP 6:?
          :? I;":";:FOR K=Ø TO
          5:N=PEEK(ADR(BUFFER$
         )+I+K-BEG):T$="ØØØ":T
         $(4-LEN(STR$(N)))=STR
         $(N)
MA 46Ø   IF PEEK(764)<255 THEN
          GET #1,A:POP :POP :?
          :GOTO 37Ø
FM 47Ø   ? T$;",";:NEXT K:? CH
         R$(126);:NEXT I:? :?
         :GOTO 37Ø
GA 48Ø   IF N<Ø THEN ? :GOTO 3
         7Ø
MM 49Ø   A(J)=N:NEXT J
JM 5ØØ   CKSUM=ADDR-INT(ADDR/2
         56)*256:FOR I=1 TO 6:
         CKSUM=CKSUM+A(I):CKSU
         M=CKSUM-256*(CKSUM>25
         5):NEXT I
KK 51Ø   RF=128:SOUND Ø,2ØØ,12
         ,8:GOSUB 57Ø:SOUND Ø,
         Ø,Ø,Ø:RF=Ø:? CHR$(126
         )
CN 52Ø   IF N<>CKSUM THEN ? :?
          "Incorrect";CHR$(253
         );:? :GOTO 37Ø
EK 53Ø   FOR W=15 TO Ø STEP -1
         :SOUND Ø,5Ø,1Ø,W:NEXT
          W
FL 54Ø   FOR I=1 TO 6:POKE ADR
         (BUFFER$)+ADDR-BEG+I-
         1,A(I):NEXT I
HB 55Ø   ADDR=ADDR+6:IF ADDR<=
         FIN THEN 37Ø
GM 56Ø   GOTO 71Ø
FI 57Ø   N=Ø:Z=Ø
PH 58Ø   GET #1,A:IF A=155 OR
         A=44 OR A=32 THEN 67Ø
FB 59Ø   IF A<32 THEN N=-A:RET
         URN
EB 6ØØ   IF A<>126 THEN 63Ø
HL 61Ø   GOSUB 69Ø:IF I=1 AND
         T=44 THEN N=-1:? CHR$
         (126);:GOTO 69Ø
GH 62Ø   GOTO 57Ø
GJ 63Ø   IF A<48 OR A>57 THEN
         58Ø
AM 64Ø   ? CHR$(A+RF);:N=N*1Ø+
         A-48
EB 65Ø   IF N>255 THEN ? CHR$(
         253);:A=126:GOTO 6ØØ
EH 66Ø   Z=Z+1:IF Z<3 THEN 58Ø
JH 67Ø   IF Z=Ø THEN ? CHR$(25
         3);:GOTO 57Ø
KC 68Ø   ? ",";:RETURN
```

```
MO 69Ø   POKE 752,1:FOR I=1 TO
          3:? CHR$(3Ø);:GET #6
         ,T:IF T<>44 AND T<>58
          THEN ? CHR$(A);:NEXT
          I
PI 7ØØ   POKE 752,Ø:? " ";CHR$
         (126);:RETURN
KM 71Ø   GRAPHICS Ø:POKE 71Ø,2
         6:POKE 712,26:POKE 7Ø
         9,2
FF 72Ø   IF MEDIA=ASC("T") THE
         N 89Ø
OJ 73Ø   REM  DISK
OK 74Ø   IF READ THEN ? :? "Lo
         ad File":?
IG 75Ø   IF DTYPE<>7Ø THEN 1Ø4
         Ø
AE 76Ø   ? :? "Enter AUTORUN.S
         YS for automatic use"
         :? :? "Enter filename
         ":INPUT T$
GF 77Ø   F$=T$:IF LEN(T$)>2 TH
         EN IF T$(1,2)<>"D:" T
         HEN F$="D:":F$(3)=T$
HJ 78Ø   TRAP 87Ø:CLOSE #2:OPE
         N #2,8-4*READ,Ø,F$:?
         :? "Working..."
JM 79Ø   IF READ THEN FOR I=1
         TO 6:GET #2,A:NEXT I:
         GOTO 82Ø
PO 8ØØ   PUT #2,255:PUT #2,255
DJ 81Ø   H=INT(BEG/256):L=BEG-
         H*256:PUT #2,L:PUT #2
         ,H:H=INT(FIN/256):L=F
         IN-H*256:PUT #2,L:PUT
          #2,H
HF 82Ø   GOSUB 97Ø:IF PEEK(195
         )>1 THEN 87Ø
IF 83Ø   IF STARTADR=Ø OR READ
          THEN 85Ø
FD 84Ø   PUT #2,224:PUT #2,2:P
         UT #2,225:PUT #2,2:H=
         INT(STARTADR/256):L=S
         TARTADR-H*256:PUT #2,
         L:PUT #2,H
HH 85Ø   TRAP 32767:CLOSE #2:?
          "Finished.":IF READ
         THEN ? :? :LET READ=Ø
         :GOTO 36Ø
HF 86Ø   END
FO 87Ø   ? "Error ";PEEK(195);
         " trying to access":?
          F$:CLOSE #2:? :GOTO
         76Ø
MC 88Ø   REM  BOOT TAPE
HH 89Ø   IF READ THEN ? :? "Re
         ad Tape"
HI 9ØØ   ? :? :? "Insert, Rewi
         nd Tape.":? "Press PL
         AY ";:IF  NOT READ TH
         EN ? "& RECORD"
LP 91Ø   ? :? "Press RETURN wh
         en ready:";
JH 92Ø   TRAP 96Ø:CLOSE #2:OPE
         N #2,8-4*READ,128,"C:
         ":? :? "Working..."
MM 93Ø   GOSUB 97Ø:IF PEEK(195
         )>1 THEN 96Ø
HH 94Ø   CLOSE #2:TRAP 32767:?
          "Finished.":? :? :IF
          READ THEN LET READ=Ø
         :GOTO 36Ø
HF 95Ø   END
CD 96Ø   ? :? "Error ";PEEK(19
         5);" when reading/wri
         ting boot tape":? :CL
         OSE #2:GOTO 89Ø
MB 97Ø   REM CIO Load/Save Fil
         e#2 opened READ=Ø fo
         r write READ=1 for r
         ead
EA 98Ø   X=32:REM File#2,$2Ø
EF 99Ø   ICCOM=834:ICBADR=836:
```

```
          ICBLEN=84Ø:ICSTAT=835
MD 1ØØØ   H=INT(ADR(BUFFER$)/2
          56):L=ADR(BUFFER$)-H
          *256:POKE ICBADR+X,L
          :POKE ICBADR+X+1,H
FH 1Ø1Ø   L=FIN-BEG+1:H=INT(L/
          256):L=L-H*256:POKE
          ICBLEN+X,L:POKE ICBL
          EN+X+1,H
MD 1Ø2Ø   POKE ICCOM+X,11-4*RE
          AD:A=USR(ADR(CIO$),X
          )
BG 1Ø3Ø   POKE 195,PEEK(ICSTAT
          ):RETURN
KA 1Ø4Ø   REM  SECTOR I/O
GC 1Ø5Ø   IF READ THEN 11ØØ
HE 1Ø6Ø   ? :? "Format Disk In
           Drive 1? (Y/N):";
FC 1Ø7Ø   GET #1,A:IF A<>78 AN
          D A<>89 THEN 1Ø7Ø
EC 1Ø8Ø   ? CHR$(A):IF A=78 TH
          EN 11ØØ
CP 1Ø9Ø   ? :? "Formatting..."
          :XIO 254,#2,Ø,Ø,"D:"
          :? "Format Complete"
          :?
AC 11ØØ   NR=INT((FIN-BEG+127)
          /128):BUFFER$(FIN-BE
          G+2)=CHR$(Ø):IF READ
           THEN ? "Reading..."
          :GOTO 112Ø
LE 111Ø   ? "Writing..."
LI 112Ø   FOR I=1 TO NR:S=I
IO 113Ø   IF READ THEN GOSUB 1
          22Ø:BUFFER$(I*128-12
          7)=SECTOR$:GOTO 116Ø
PL 114Ø   SECTOR$=BUFFER$(I*12
          8-127)
AM 115Ø   GOSUB 122Ø
DM 116Ø   IF PEEK(DSTATS)<>1 T
          HEN 12ØØ
FB 117Ø   NEXT I
GM 118Ø   IF  NOT READ THEN EN
          D
DH 119Ø   ? :? :LET READ=Ø:GOT
          O 36Ø
JJ 12ØØ   ? "Error on disk acc
          ess.":? "May need fo
          rmatting.":GOTO 1Ø4Ø
KI 121Ø   REM
BL 122Ø   REM  SECTOR ACCESS S
          UBROUTINE
IG 123Ø   REM Drive ONE
IH 124Ø   REM Pass buffer in S
          ECTOR$
MP 125Ø   REM sector # in vari
          able S
EG 126Ø   REM READ=1 for read,
KJ 127Ø   REM READ=Ø for write
BM 128Ø   BASE=3*256
GL 129Ø   DUNIT=BASE+1:DCOMND=
          BASE+2:DSTATS=BASE+3
HL 13ØØ   DBUFLO=BASE+4:DBUFHI
          =BASE+5
AI 131Ø   DBYTLO=BASE+8:DBYTHI
          =BASE+9
JA 132Ø   DAUX1=BASE+1Ø:DAUX2=
          BASE+11
PM 133Ø   REM DIM DSKINV$(4)
CA 134Ø   DSKINV$="hLS":DSKINV
          $(4)=CHR$(228)
PF 135Ø   POKE DUNIT,1:A=ADR(S
          ECTOR$):H=INT(A/256)
          :L=A-256*H
BP 136Ø   POKE DBUFHI,H
CD 137Ø   POKE DBUFLO,L
PD 138Ø   POKE DCOMND,87-5*REA
          D
AA 139Ø   POKE DAUX2,INT(S/256
          ):POKE DAUX1,S-PEEK(
          DAUX2)*256
KJ 14ØØ   A=USR(ADR(DSKINV$))
KG 141Ø   RETURN
```

64 MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX
   {SPACE}VERSION 2.00 ARE 750
   ,765,770 AND 860    :rem 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 :rem 147
100 PRINT"{CLR}{6}";CHR$(142);
    CHR$(8);:POKE53281,1:POKE5
    3280,1              :rem 67
101 POKE 788,52:REM DISABLE RU
    N/STOP             :rem 119
110 PRINT"{RVS}{39 SPACES}";
                       :rem 176
120 PRINT"{RVS}{14 SPACES}
    {RIGHT}{OFF}{*}£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}
    {*}{OFF}{*}£{RVS}£{RVS}
    {14 SPACES}";      :rem 250
130 PRINT"{RVS}{14 SPACES}
    {RIGHT} {G}{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£
    {*}{OFF}{*}{RVS}
    {14 SPACES}";       :rem 35
140 PRINT"{RVS}{41 SPACES}"
                       :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} M
    ACHINE LANGUAGE EDITOR VER
    SION 2.02{5 DOWN}":rem 238
210 PRINT"{5}{2 UP}STARTING AD
    DRESS?{8 SPACES}{9 LEFT}";
                       :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+11
    9*F)               :rem 166
220 IFS<256OR(S>40960ANDS<4915
    2)ORS>53247THENGOSUB3000:G
    OTO210             :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"{5}{2 UP}ENDING ADDR
    ESS?{8 SPACES}{9 LEFT}";:I
    NPUTE:F=1-F:C$=CHR$(31+119
    *F)                 :rem 20
240 IFE<256OR(E>40960ANDE<4915
    2)ORE>53247THENGOSUB3000:G
    OTO230             :rem 183
250 IFE<STHENPRINTC$;"{RVS}END
    ING < START{2 SPACES}":GOS
    UB1000:GOTO 230   :rem 176
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S
                        :rem 56
310 A=1:PRINTRIGHT$("0000"+MID
    $(STR$(AD),2),5);":";
                        :rem 33
315 FORJ=ATO6           :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:G
    OTO320             :rem 228
390 IFN=-211THEN 710    :rem 62
400 IFN=-204THEN 790    :rem 64
410 IFN=-206THENPRINT:INPUT"
    {DOWN}ENTER NEW ADDRESS";Z
    Z                   :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETH
    ENPRINT"{RVS}OUT OF RANGE"
    :GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GO
    TO310              :rem 238
420 IF N<>-196 THEN 480
                       :rem 133
430 PRINT:INPUT"DISPLAY:FROM";
    F:PRINT,"TO";:INPUTT
                       :rem 234
440 IFF<SORF>EORT<SORT>ETHENPR
    INT"AT LEAST";S;"{LEFT}, N
    OT MORE THAN";E:GOTO430
                       :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINT
    RIGHT$("0000"+MID$(STR$(I)
    ,2),5);":";         :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRIN
    TRIGHT$("00"+MID$(STR$(N),
    2),3);",";          :rem 66
460 GETA$:IFA$>""THENPRINT:PRI
    NT:GOTO310          :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI
    :PRINT:PRINT:GOTO310
                        :rem 50
480 IFN<0 THEN PRINT:GOTO310
                       :rem 168
490 A(J)=N:NEXTJ       :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
    ORI=1TO6:CKSUM=(CKSUM+A(I)
    )AND255:NEXT       :rem 200
510 PRINTCHR$(18);:GOSUB570:PR
    INTCHR$(146);       :rem 94
511 IFN=-1THENA=6:GOTO315
                       :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHE
    N530               :rem 122
520 PRINT:PRINT"LINE ENTERED W
    RONG : RE-ENTER":PRINT:GOS
    UB1000:GOTO310     :rem 176
530 GOSUB2000          :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):
    NEXT:POKE54272,0:POKE54273
    ,0                 :rem 227
550 AD=AD+6:IF AD<E THEN 310
                       :rem 212
560 GOTO 710           :rem 108
570 N=0:Z=0             :rem 88
580 PRINT"{£}";         :rem 81
581 GETA$:IFA$=""THEN581
                        :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*
    (A$=".")-4*(A$="J")-5*(A$=
    "K")-6*(A$="L")     :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I"
    )-9*(A$="O"):IFA$="H"THENA
    $="0"              :rem 134
584 IFAV>0THENA$=CHR$(48+AV)
                       :rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
    FA=13ORA=44ORA=32THEN670
                       :rem 229
590 IFA>128THENN=-A:RETURN
                       :rem 137
600 IFA<>20 THEN 630    :rem 10
610 GOSUB690:IFI=1ANDT=44THENN
    =-1:PRINT"{OFF}{LEFT}
    {LEFT}";:GOTO690    :rem 62
620 GOTO570            :rem 109
630 IFA<48ORA>57THEN580
                       :rem 105
640 PRINTA$;:N=N*10+A-48
                       :rem 106
650 IFN>255 THEN A=20:GOSUB100
    0:GOTO600          :rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
                       :rem 114
680 PRINT",";:RETURN   :rem 240
690 S%=PEEK(209)+256*PEEK(210)
    +PEEK(211)         :rem 149
691 FORI=1TO3:T=PEEK(S%-I)
                        :rem 67
695 IFT<>44ANDT<>58THENPOKES%-
    I,32:NEXT          :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
    ;:RETURN             :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *
    **{3 DOWN}"        :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L SAVE){DOWN}"     :rem 106
720 F$="":INPUT"{DOWN} FILENAM
    E";F$:IFF$=""THENPRINT:PRI
    NT:GOTO310          :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"           :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN740              :rem 36
750 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$:OPEN15,8,15,"S"
    +F$:CLOSE15        :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                    :rem 3
762 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                       :rem 109
763 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466       :rem 69
765 K=S:POKE254,K/256:POKE253,
    K-PEEK(254)*256:POKE780,25
    3                   :rem 17
766 K=E+1:POKE782,K/256:POKE78
    1,K-PEEK(782)*256:SYS65496
                       :rem 235
770 IF(PEEK(783)AND1)OR(191AND
    ST)THEN780         :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":G
    OTO310             :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.
    {2 SPACES}TRY AGAIN.":IFDV
    =1THEN720          :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO720              :rem 103
790 PRINT"{CLR}{RVS}*** LOAD *
    **{2 DOWN}"        :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L LOAD)"            :rem 82
800 F$="":INPUT"{2 DOWN} FILEN
    AME";F$:IFF$=""THENPRINT:G
    OTO310             :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"           :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN820              :rem 34
830 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$         :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                    :rem 2
841 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                       :rem 107
845 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466       :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191AND
    ST)THEN870         :rem 111
865 PRINT"{DOWN}DONE.":GOTO310
                        :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.
    {2 SPACES}TRY AGAIN.{DOWN}
    ":IFDV=1THEN800    :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO800              :rem 102
1000 REM BUZZER        :rem 135
1001 POKE54296,15:POKE54277,45
     :POKE54278,165    :rem 207
1002 POKE54276,33:POKE 54273,6
     :POKE54272,5       :rem 42
1003 FORT=1TO200:NEXT:POKE5427
     6,32:POKE54273,0:POKE5427
     2,0:RETURN        :rem 202
2000 REM BELL SOUND     :rem 78
2001 POKE54296,15:POKE54277,0:
     POKE54278,247     :rem 152
2002 POKE 54276,17:POKE54273,4
     0:POKE54272,0      :rem 86
2003 FORT=1TO100:NEXT:POKE5427
     6,16:RETURN        :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PA
     GE OR ROM":GOTO1000
                        :rem 89
```

©

Saving Time And Memory:
An Atari Variable Utility

P. E. Thompson

Here's a utility—actually three separate programs—which can help programmers save time and conserve memory. With them, you can list, rename, and abbreviate all variable names in a BASIC program. A thorough explanation is included.

One valuable feature of Atari BASIC is its provision for long variable names—up to 128 characters, with every character significant. Naming variables for what they represent, such as AVERAGE, rather than using a cryptic code, like A, makes programs self-documenting and more readable.

However, there are two disadvantages. First, if you want to rename a variable, it is time-consuming to go back through an entire program to edit long variable names. Second, long names lengthen program lines and make it difficult to add statements to the lines later. (Long variable names, however, don't consume much more memory; the Atari stores every character of a name only for the first reference, and uses a lookup table for subsequent references.)

The utility programs following this article solve both problems. In addition, the program steps are explained in detail so you can understand what's happening. If you wish, you can readily modify the programs or use some of the same techniques in your own programming.

The Variable Name Table

Changing variable names in Atari BASIC is actually very easy. Each name is stored in a lookup table called the Variable Name Table. When a program is being listed, BASIC references this table each time a variable appears. When you change a name in the table, every name in the program listing also changes.

You can locate the Variable Name Table by examining memory locations 130 and 131 (decimal) for the start of the table, and locations 132 and 133 for the end of the table. Try this example. Load a BASIC program, type the following line in immediate mode (no line number), and press RETURN:

```
FOR X=PEEK(130)+PEEK(131)*256
  TO PEEK(132)+PEEK(133)*256:
  PRINT CHR$(PEEK(X));:NEXT X
```

This line converts the bytes in those addresses to decimal locations by adding the least significant byte (LSB) to the product of the most significant byte (MSB) times 256. Then it displays the character representations of each memory position between those locations. These character representations are the Variable Name Table.

The table does not look quite as you might expect. Sprinkled throughout are characters in inverse video. These characters are flags which signal the end of a variable name and indicate the variable type. If the type is a scalar variable (that is, a number), the last character of the name is in inverse video. For string variables, an inverse-video dollar sign is appended. For an array variable, an inverse-video left parenthesis is added.

By scanning the table, you may see variable names that no longer appear in the program itself. This can happen for two reasons. First, mistyped commands entered in immediate mode while you're programming may be inadvertently interpreted by BASIC as variable names, and therefore added to the table. Second, variable names used in a program but later removed are not deleted from the name table.

The only way to remove these unused names is to LIST the program to tape or disk, type NEW to erase the program in memory, and then re-ENTER the program. When you load a program with ENTER, BASIC reinterprets each line as if you were typing the program manually. (That's why ENTER takes longer than LOAD.)

Using The Utilities

Follow these steps to use each utility:

1. Type each one into the computer individually from the listings here. REM lines are included strictly for reference and can be eliminated to save typing.

2. Store each utility on tape or disk using the LIST command, not SAVE.

3. Type NEW to erase any program in memory. Load the program on which the utility will operate. Make sure the program has no line numbers greater than 31999.

4. Load the appropriate utility using the ENTER command. For example, ENTER"C:" for tape or ENTER"D:*filename*" for disk. This appends the utility to the end of the program. (If your program has line numbers greater than 31999, they will be replaced by the utility.)

5. Run the utility by typing GOTO 32000 and pressing RETURN.

6. Write down the two starting addresses of the Variable Name Table. If a utility has run but an error has been made or a change is required, these addresses must be restored before any computer operations can take place. To restore the addresses, POKE 130 with the location 130 value listed by the utility, and POKE 131 with the location 131 value listed by the utility.

7. Execute the utility by responding to the screen prompts.

8. Two of the utilities—"Changer" and "Squeezer"—require that you immediately save the newly modified version of your program on tape or disk. However, you can't use the SAVE command for this purpose because the utility is merged with your program, so both would be saved together. Nor can you save the program with an immediate mode command, because the Variable Name Table would become garbled. Therefore, line 32380 in Changer and Squeezer automatically LISTs the modified program to tape or disk, separating it from the utility in the process. The utilities currently are set up to LIST your program to disk with the filename D:XXXXXXXX. XXX. You can change this filename by modifying line 32380 in both Changer and Squeezer. Also, change line 32380 in both utilities to LIST"C:",0,31999 for cassette.

9. After Changer or Squeezer has automatically saved your program, clear the computer by turning it off, then on again. Then you can load your program with the ENTER command for a test run. This assures that all pointers and the Variable Name Table will be reset to proper values.

Lister

The first utility, "Lister," lists the variable names and types. It scans the Variable Name Table looking for inverse characters to determine the type of variable. Each variable and its type are listed in the order of appearance in the table. More specific descriptions of the utility's steps are included in the program listing.

If you want hardcopy, change the PRINT statements in lines 32040, 32140, 32160, and 32180 to LPRINT.

Changer

The second utility, "Changer," displays each variable on the screen and gives you the opportunity to change it. Press RETURN to retain the variable name.

Changer operates by adding either the existing name or the changed name to a string variable called VARNAME$. This string emulates the format of the Variable Name Table, including the inverse video flags. When you've been given a chance to change all the names, Changer makes VARNAME$ the new name table. It does this by finding the starting memory location of VARNAME$ with the ADR function, then computing revised values for locations 130 and 131 and POKEing them into place.

Immediately after Changer has LISTed your program to disk or tape, reboot the computer as described in step 9.

You may want to expand the size of the new Variable Name Table. A program using many variables or long names may have insufficient space dimensioned for the new name table. If all the space in the new table is used before the utility has completed, an Error 5, String Length Error, will result. To allocate more space, change the dimensioned value for VARNAME$ in line 32020 from 500 to a larger number. You'll have to use your judgment as to the size of the number based on the number of variables and the length of the names.

Squeezer

The third utility is "Squeezer." It is similar to Changer except that each variable name is automatically replaced by a unique one- or two-letter name. This shortens the Variable Name Table to its minimum length, yet preserves the ability to LIST or modify the program. It's intended for use after a program is completely developed and debugged, particularly when the program requires as much free memory as possible. It's also helpful for shortening long program lines so you can add more statements. During testing, Squeezer reduced the size of one program by 400 bytes—an impressive figure, especially if you're working on a 16K computer.

Squeezer lists the variable type, original name, and revised name. If you want a hardcopy, add the following line:

```
32001 OPEN #1,8,0,"P:"
```

and change the PRINT statements in lines 32045, 32050, 32060, 32160, 32181, 32201, 32220, 32260, and 32300 to PRINT #1;.

As with Changer, after Squeezer has LISTed your program on disk or tape, immediately reboot the computer as described in step 9.

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: Lister

```
FD 32ØØØ PRINT CHR$(125):? :
         ?
GF 32Ø1Ø REM INITIALIZE VARI
         ABLES
NE 32Ø11 REM NAME$=VARIABLE
         NAME
FB 32Ø12 REM LOCATION=MEMORY
          ADDRESS
BN 32Ø2Ø CLR :DIM NAME$(128)
NP 32Ø3Ø GOSUB 32Ø4Ø:GOTO 32
         Ø6Ø
BJ 32Ø4Ø NAME$="":? "Type  :
          Variable Name":RET
         URN
HA 32Ø5Ø REM BEGIN FOR-NEXT
         LOOP
NI 32Ø51 REM FROM STARTING L
         OCATION
EE 32Ø52 REM OF VARIABLE NAM
         E TABLE
JC 32Ø53 REM TO ENDING LOCAT
         ION
IL 32Ø6Ø FOR LOCATION=PEEK(1
         3Ø)+PEEK(131)*256 T
         O PEEK(132)+PEEK(13
         3)*256-1
IN 32Ø7Ø REM CHECK FOR INVER
         SE CHAR.
PB 32Ø71 REM IF NOT,ADD TO N
         AME STRING
CN 32Ø72 REM AND GET NEXT LO
         CATION
PI 32Ø8Ø IF PEEK(LOCATION)<1
         28 THEN NAME$(LEN(N
         AME$)+1)=CHR$(PEEK(
         LOCATION)):NEXT LOC
         ATION
DN 32Ø9Ø REM IF LOCATION IS
         NOT A $
KD 32Ø91 REM THEN JUMP AHEAD
JE 321ØØ IF PEEK(LOCATION)<>
         164 THEN 3216Ø
LB 3211Ø REM IF VARIABLE IS
         "NAME"
CB 32111 REM VARIABLES IN TH
         E UTILITY
FH 32112 REM HAVE BEEN ENCOU
         NTERED
BK 32113 REM SO WE ARE DONE
HK 3212Ø REM IF NAME$="NAME"
          THEN 3222Ø
KF 3213Ø REM SINCE LAST CHAR
         ACTER OF
OP 32131 REM THE NAME IS $ P
         RINT TYPE
PN 32132 REM "STRING" AND TH
         E NAME.
FI 32133 REM GET NEXT LOCATI
         ON
MG 3214Ø PRINT "STRING: ";NA
         ME$:GOTO 322ØØ
BC 3215Ø REM SINCE LAST CHAR
         ACTER
LL 32151 REM OF THE NAME IS
         (
FD 32152 REM PRINT "ARRAY" A
         ND NAME.
FK 32153 REM GET NEXT LOCATI
         ON
IL 3216Ø IF PEEK(LOCATION)=1
         68 THEN ? "ARRAY: "
         ;NAME$:GOTO 322ØØ
BE 3217Ø REM SINCE LAST CHAR
         ACTER
HN 32171 REM OF NAME IS INVE
         RSE,
CD 32172 REM CHANGE TO NORMA
         L.
IN 32173 REM PRINT "SCALAR"
         AND NAME.
IL 32174 REM GET NEXT LOCATI
         ON.
LI 3218Ø NAME$(LEN(NAME$)+1)
         =CHR$(PEEK(LOCATION
         )-128):? "SCALAR: "
         ;NAME$
CN 3219Ø REM IF SCREEN IS FU
         LL,
NF 32191 REM STOP AND WAIT F
         OR INPUT,
CI 32192 REM RESET SCREEN
KC 32193 REM FOR MORE NAMES.
AI 322ØØ IF PEEK(84)>2Ø THEN
          ? :? "PRESS RETURN
          TO CONTINUE";:INPU
         T NAME$:? CHR$(125)
         :GOSUB 32Ø4Ø
KE 3221Ø REM RESET NAME$
HH 32211 REM FOR NEXT VARIAB
         LE.
IE 32212 REM GET NEXT LOCATI
         ON.
JB 3222Ø NAME$="":NEXT LOCAT
         ION
NC 3224Ø END
```

Program 2: Changer

```
BA 32ØØØ ? CHR$(125):? :?
GF 32Ø1Ø REM INITIALIZE VARI
         ABLES
LH 32Ø11 REM ZNAME$  =OLD NA
         ME
LK 32Ø12 REM VARNAME$=NEW NA
         ME TABLE
AB 32Ø13 REM RENAME$ =NEW NA
         ME
FD 32Ø14 REM LOCATION=MEMORY
          ADDRESS
KL 32Ø2Ø CLR :DIM ZNAME$(128
         ),VARNAME$(5ØØ),REN
         AME$(128)
LA 32Ø22 ? "VALUE AT LOCATIO
         N 13Ø: ";PEEK(13Ø):
         ? "VALUE AT LOCATIO
         N 131: ";PEEK(131):
         ?
FI 32Ø3Ø GOSUB 32Ø4Ø:? :GOTO
          32Ø6Ø
HD 32Ø4Ø ZNAME$="":? "Type
         : Variable Name":RE
         TURN
NA 32Ø5Ø REM BEGIN FOR-NEXT
         LOOP
NI 32Ø51 REM FROM STARTING L
         OCATION
EE 32Ø52 REM OF VARIABLE NAM
         E TABLE
JC 32Ø53 REM TO ENDING LOCAT
         ION
IL 32Ø6Ø FOR LOCATION=PEEK(1
         3Ø)+PEEK(131)*256 T
         O PEEK(132)+PEEK(13
         3)*256-1
IN 32Ø7Ø REM CHECK FOR INVER
         SE CHAR.
PB 32Ø71 REM IF NOT,ADD TO N
         AME STRING
CN 32Ø72 REM AND GET NEXT LO
         CATION
KM 32Ø8Ø IF PEEK(LOCATION)<1
         28 THEN ZNAME$(LEN(
         ZNAME$)+1)=CHR$(PEE
         K(LOCATION)):NEXT L
         OCATION
PL 32Ø9Ø REM IF LOCATION IS
         NOT $
KB 32Ø91 REM THEN JUMP AHEAD
MN 321ØØ IF PEEK(LOCATION)<>
         164 THEN GOTO 3216Ø
LB 3211Ø REM IF VARIABLE IS
         "NAME"
AE 32111 REM VARIABLES IN CH
         ANGER
FH 32112 REM HAVE BEEN ENCOU
         NTERED
BK 32113 REM SO WE ARE DONE
IG 3212Ø IF ZNAME$="ZNAME" T
         HEN GOTO 3234Ø
BA 3213Ø REM SINCE LAST CHAR
         ACTER
NE 32131 REM OF NAME IS $
KJ 32132 REM PRINT "STRING"
         AND NAME.
FI 32133 REM GET NEXT LOCATI
         ON
NC 3214Ø ? "STRING: ";ZNAME$
         :GOTO 322ØØ
BC 3215Ø REM SINCE LAST CHAR
         ACTER
NK 32151 REM OF NAME IS (
FD 32152 REM PRINT "ARRAY" A
         ND NAME.
FK 32153 REM GET NEXT LOCATI
         ON
OF 3216Ø IF PEEK(LOCATION)=1
         68 THEN ? "ARRAY :
         ";ZNAME$:GOTO 322ØØ
BE 3217Ø REM SINCE LAST CHAR
         ACTER
NG 32171 REM OF ZNAME IS INV
         ERSE,
CD 32172 REM CHANGE TO NORMA
         L.
IN 32173 REM PRINT "SCALAR"
         AND NAME.
FN 32174 REM GET NEXT LOCATI
         ON
MG 3218Ø ZNAME$(LEN(ZNAME$)+
         1)=CHR$(PEEK(LOCATI
         ON)-128):? "SCALAR:
          ";ZNAME$
PP 3219Ø REM INPUT NEW NAME
         OR RETURN
LG 32191 REM IF NO CHANGE
CF 322ØØ ? :? "NEW NAME OR R
         ETURN":INPUT RENAME
         $
MN 3221Ø REM USE DOWN-ARROW
         TO SLIDE
JJ 32211 REM NAME OFF SCREEN
AD 3222Ø POSITION Ø,7:FOR LI
         NE=1 TO 15:? CHR$(1
         57):NEXT LINE:POSIT
         ION 2,7
IP 3223Ø REM IF RETURN PRESS
         ED,
JN 32231 REM ADD OLD NAME TO
          NEW TABLE
DK 3224Ø IF LEN(RENAME$)=Ø T
         HEN RENAME$=ZNAME$
NA 3225Ø REM IF VARIABLE IS
         ARRAY
AP 32251 REM OR STRING ADD (
          OR $
AN 3226Ø IF PEEK(LOCATION)=1
         64 OR PEEK(LOCATION
         )=168 THEN RENAME$(
         LEN(RENAME$)+1)=CHR
         $(PEEK(LOCATION)):G
         OTO 323ØØ
AJ 3227Ø REM IF VARIABLE IS
         SCALAR
```

```
NL 32271 REM CHANGE LAST CHA
         R
KD 32272 REM TO INVERSE
FJ 322B0 RENAME$(LEN(RENAME$
         ))=CHR$(ASC(RENAME$
         (LEN(RENAME$)))+12B
         ):GDTD 32300
FL 32290 REM ADD NAME TO NEW
LE 32291 REM VARIABLE NAME T
         ABLE
EK 32300 VARNAME$(LEN(VARNAM
         E$)+1)=RENAME$
PP 32310 REM RESET ZNAME$
HI 32311 REM FOR NEXT VARIAB
         LE.
HC 32312 REM GET NEXT VARIAB
         LE.
ID 32320 ZNAME$="":RENAME$="
         ":NEXT LOCATION
HC 32330 REM ALL VARIABLE NA
         MES
AO 32331 REM REVISEO. AOO CH
         R$(0) TD
AE 32332 REM TABLE TO--INDICA
         TE END
PE 32340 VARNAME$(LEN(VARNAM
         E$)+1)=CHR$(0)
EE 32350 REM CHANGE ORIGINAL
          TABLE
NN 32351 REM AOORESS TD NEW
         TABLE
MK 32360 POKE 131,INT(ADR(VA
         RNAME$)/256):POKE 1
         30,AOR(VARNAME$)-PE
         EK(131)*256
AC 32370 ? CHR$(125):? "NDW
         LISTING TO TAPE OR
         DISK.":? "CHANGE LI
         NE 323B0 IF DESIRED
         ."
IL 323B0 LIST "O:XXXXXXXX.XX
         X",0,31999
NI 32390 ENO
```

Program 3: Squeezer

```
BA 32000 ? CHR$(125):? :?
GK 32011 REM COUNT(0)= NUM.
         STRINGS
BE 32012 REM CDUNT(1)= NUM.
         ARRAYS
EN 32013 REM COUNT(2)= NUM.
         SCALARS
EI 32014 REM COUNT(3)= COUNT
         ER
AD 32015 REM COUNT(4)= ARGUM
         ENT IN SUB
LO 32016 REM VARNAME$= NEW N
         AME TABLE
EM 32019 CLR :OIM VARNAME$(3
         B4),COUNT(4)
KO 32020 ? "VALUE AT LOCATIO
         N 130: ";PEEK(130):
         ? "VALUE AT LDCATIO
         N 131: ";PEEK(131):
         ?
FM 32022 COUNT(0)=0:COUNT(1)
         =0:COUNT(2)=0:COUNT
         (3)=0:COUNT(4)=0:GO
         SUB 32040:GOTO 3212
         0
GP 32030 REM SUBROUTINES TO
         PRINT
JH 32031 REM VARIABLE NAMES
HA 32040 IF PEEK(B4)<22 THEN
          GOTO 32045
PC 32041 ? "PRESS RETURN TO
         CONTINUE"
OB 32042 IF PEEK(764)<>12 TH
         EN GOTO 32042
PG 32043 POKE 764,155:? CHR$
         (125)
DB 32045 ? "  NAME: ";:RETUR
         N
ME 32050 ? "RENAME: ";:RETUR
         N
GM 32060 ? VARNAME$(LEN(VARN
         AME$));:RETURN
DA 32070 REM SUBRDUTINE TO O
         ETERMINE
OP 32071 REM NEW VARIABLE NA
         ME.  IF
MB 32072 REM ALL SINGLE LETT
         ER NAMES
HO 32073 REM HAVE BEEN USED,
HK 32074 REM ADD A SECONO LE
         TTER
FH 320B0 GDSUB 32050:IF COUN
         T(4)<25 THEN GDTO 3
         2090
OB 320B5 CDUNT(3)=1+INT(COUN
         T(4)/25):VARNAME$(L
         EN(VARNAME$)+1)=CHR
         $(64+COUNT(3)):GOSU
         B 32060
PE 32090 COUNT(3)=1+COUNT(4)
         -INT(CDUNT(4)/25)*2
         5:VARNAME$(LEN(VARN
         AME$)+1)=CHR$(64+CD
         UNT(3))
IL 32100 GOSUB 32060:RETURN
LO 32110 REM CHECK ALL LOCAT
         IONS
BI 32111 REM FRDM START TO E
         ND
PL 32112 REM OF NAME TABLE
CK 32120 FOR LOCATIDN=PEEK(1
         30)+PEEK(131)*256 T
         D PEEK(132)+PEEK(13
         3)*256
EG 32130 REM IF CHARACTER IS
          CHR$(0) THEN
OK 32131 REM ENO OF TABLE IS
          REACHED
CF 32140 IF PEEK(LDCATION)=0
          THEN GDTD 32300
II 32150 REM IF CHARACTER IS
          NDT
CM 32151 REM INVERSE THEN GE
         T NEXT ONE
JC 32152 REM IF INVERSE THEN
          ENO
MC 32153 REM OF NAME IS REAC
         HEO SD
AI 32154 REM OETERMINE VARIA
         BLE TYPE
GM 32160 IF PEEK(LOCATION)<1
         27 THEN ? CHR$(PEEK
         (LOCATION));:GOTO 3
         22B0
HA 32170 REM IF CHARACTER IS
          ( THEN
FJ 32171 REM TYPE IS ARRAY.
         SET
LK 32172 REM ARGUMENT TO COU
         NT, CALL
DE 32173 REM SUBROUTINE TO O
         ETERMINE
IO 32174 REM VARIABLE NAME.
         AOO ( TO
IF 32175 REM NAME, AOO 1 TO
         COUNT,
CH 32176 REM GET NEXT NAME
JL 321B0 IF PEEK(LOCATION)<>
         16B THEN 32200
KK 321B1 ? "("
DH 321B2 COUNT(4)=COUNT(1):G
         OSUB 320B0:VARNAME$
         (LEN(VARNAME$)+1)="
         (":GOSUB 32060:COUN
         T(1)=COUNT(1)+1:GOT
         O 32260
PP 32190 REM IF CHAR IS $ TH
         EN
LO 32191 REM TYPE IS STRING.
          SET
LM 32192 REM ARGUMENT TD CDU
         NT, CALL
DI 32195 REM SUBROUTINE TO O
         ETERMINE
IO 32196 REM VARIABLE NAME.
         ADD $ TO
IJ 32197 REM NAME, ADD 1 TD
         COUNT,
CL 3219B REM GET NEXT NAME
ML 32200 IF PEEK(LOCATION)<>
         164 THEN GOTO 32220
JP 32201 ? "$"
CJ 32202 CDUNT(4)=COUNT(0):G
         DSUB 320B0:VARNAME$
         (LEN(VARNAME$)+1)="
         $":GOSUB 32060:COUN
         T(0)=COUNT(0)+1:GOT
         D 32260
DE 32210 REM VARIABLE TYPE I
         S SCALAR.
FB 32211 REM PRINT NORMAL CH
         AR
CB 32220 ? CHR$(PEEK(LDCATIO
         N)-12B)
DI 32230 REM SET ARGUMENT ED
         UAL TD NUM
DP 32231 REM OF SCALAR VARIA
         BLES FOUNO
LF 32232 REM SO FAR. CALL SU
         BRDUTINE
FK 32233 REM TO OETERMINE NE
         W NAME.
FB 32234 REM ADO 1 TO NUMBER
          SCALARS
PE 32240 CDUNT(4)=COUNT(2):G
         OSUB 320B0:COUNT(2)
         =CDUNT(2)+1
CC 32250 REM SET LAST CHARAC
         TER OF
MB 32251 REM NAME TO INVERSE
NJ 32260 VARNAME$(LEN(VARNAM
         E$))=CHR$(ASC(VARNA
         ME$(LEN(VARNAME$)))
         +12B):? :? :GOSUB 3
         2040
NL 32270 REM END OF FOR-NEXT
          LDOP
FF 32271 REM FOR NEXT CHAR.
JH 322B0 NEXT LOCATION
AM 32290 REM HOLO LAST PARTI
         AL SCREEN
BA 32291 REM FOR OISPLAY.
EL 32292 REM AOO CHR$(0) TO
         ENO OF NEW NAME
CB 32293 REM NAME TABLE INOI
         CATING ENO
KD 32300 ? "ENO OF TABLE":?
         :GOSUB 32041:VARNAM
         E$(LEN(VARNAME$)+1)
         =CHR$(0)
PO 32330 REM CHANGE TABLE AO
         ORESS
BL 32340 POKE 131,INT(AOR(VA
         RNAME$)/256):POKE 1
         30,AOR(VARNAME$)-IN
         T(AOR(VARNAME$)/256
         )*256
BC 32350 REM OISPLAY WARNING
          MESSAGE
NO 32360 ? CHR$(125):? "NOW
         LISTING TO TAPE OR
         OISK":? "CHANGE LIN
         E 323B0 IF OESIREO.
         "
IL 323B0 LIST "O:XXXXXXXX.XX
         X",0,31999
NI 32390 ENO                ©
```

Commodore 64 Disk Commander

Michael Kunkel

*Disk access can be clumsy on the Commodore 64 because it has no special disk commands like those found on the Commodore Plus/4, 16, and PET/CBM computers. "Disk Commander" is a powerful new utility which adds the missing commands, plus a few more. It works with any 1541-compatible disk drive. Together with "TurboDisk" (*COMPUTE!*, April 1985), it transforms your 64 into a much faster and friendlier computer.*

Because the Commodore 64 contains BASIC 2.0, designed primarily for cassette storage, disk access is a little inconvenient. For instance, you have to type LOAD"$",8 and LIST to view a disk directory—thereby wiping out a resident BASIC program—or OPEN15,8,15,"S0:*filename*": CLOSE15 just to scratch a file. If you merely want to check the disk drive error channel, you have to write a short BASIC program. Other disk operations are equally awkward. Quite a few 64 users have pined for the more powerful BASIC 3.5 or 4.0 found in some other Commodore computers. Now that wish can come true.

"Commodore 64 Disk Commander" adds 18 commands to BASIC to simplify use of the 1541 disk drive. Furthermore, the commands are flexible enough to be included within BASIC programs, and some of the commands can't be found even in BASIC 4.0. In addition, Disk Commander resides in the Random Access Memory (RAM) hidden beneath the Commodore 64's Read Only Memory (ROM), so it's relatively protected from interference with other BASIC and machine language programs. In fact, nearly all of the commands are compatible with "TurboDisk," the high-speed disk loader published in the April 1985 issue of COMPUTE!.

Typing The Program

Disk Commander is easy to prepare. Type it in with the MLX machine language entry program found elsewhere in this issue. MLX makes it easier to type machine language programs without errors because it detects most typos after you enter each program line. (See instructions in the MLX article.)

Before using MLX to enter the data for Disk Commander, clear the computer by turning it off, then on again. Then enter the following line and press RETURN:

```
POKE 44,20:POKE 5120,0:NEW
```

Now load and run MLX. Enter these responses to the prompts:

```
Starting Address? 2049
Ending Address?  4760
```

When you're done typing, MLX automatically prompts you to save the program. You can also enter the listing in multiple sittings by following the instructions in the MLX article. If you do enter the listing in more than one sitting, remember to reset the computer and enter the above POKEs and NEW each time before loading the MLX program.

Once you've saved a copy of Disk Commander, load and run it like any BASIC program. (The POKEs are *not* necessary to run the finished program.) It will copy itself into a safe place in memory and then delete its loader program from memory. Once Disk Commander is activated, even pressing RUN/STOP–RESTORE for a warm-start reset will not disable it. Disk Commander can be turned off only by a cold-start reset (shutting off the computer or typing SYS 64738).

Command Summary

Following is a list of the new commands added by Disk Commander. Each command can be abbreviated as shown in the parentheses.

DIRECTORY (DI SHIFT-R) Calls up a disk directory without erasing a resident BASIC program.

DISKST (DI SHIFT-S) Prints the error message from the disk drive error channel.

DSAVE *"filename"* (D SHIFT-S) Saves a BASIC or machine language program with the specified filename.

DLOAD *"filename"* (D SHIFT-L) Loads a BASIC or machine language program with the specified filename.

DVERIFY *"filename"* (D SHIFT-V) Compares the program specified by the filename with the program in memory.

SCRATCH *"filename"* (S SHIFT-C) Deletes the specified file from the disk. First it asks, ARE YOU SURE? If you respond by typing YES or Y, the file is scratched.

RENAME *"oldfile"* TO *"newfile"* (RE SHIFT-N) Changes the filename from *oldfile* to *newfile*.

COPY *"file1"* TO *"file2"* (CO SHIFT-P) Makes a copy of *file1* as *file2* on the same disk. However, it does not allow you to copy a file from one disk to another.

COLLECT (CO SHIFT-L) Validates the disk by reconstructing the Block Allocation Map as explained in the disk drive manual (equivalent to OPEN 15,8,15: PRINT#15,"V0:": CLOSE 15).

HEADER *"diskname,ID"* (HE SHIFT-A) Formats a disk as described in the disk drive manual. (HEADER corresponds to the disk NEW command.) The disk is given the title *diskname* for directory purposes, and the ID should be a unique two-character combination. Any files currently on the disk will be erased when this command is executed.

DOPEN#*x*,*"filename"* (D SHIFT-O) Opens a file to the disk drive as specified by *x* and the *filename*. The filename can also specify the type (P for program, S for sequential, or L and the record length for relative files) and whether the file is being opened for reading (R) or writing (W). If these parameters are not specified, certain default values are assumed. For example, DOPEN#1, "TEST" opens file 1 for reading if TEST is an existing sequential or program file, and for both reading and writing if TEST is an existing relative file. Examples: DOPEN#1, "TEST,W" opens the sequential file TEST for writing. DOPEN#1, "TEST,P,R" opens the program file TEST for reading. DOPEN#1, "TEST,L20" creates a relative file with the filename TEST and a record length of 20. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use D SHIFT-O 1,"TEST".)

APPEND#*x*,*"filename"* (A SHIFT-P) Allows you to add data to an existing sequential file. The specified file *x* is opened for the sequential file specified by *filename*. Any data written to file *x* will be added at the end of the existing sequential file. Example: APPEND#1,"TEST": PRINT#1,"NAME": CLOSE1. This command is only for sequential files; it cannot be used to append lines to a program file. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use A SHIFT-P 1,"TEST".)

RECORD#*x*,*y*,*z* (RE SHIFT-C) Selects record *y* and character *z* in the relative file currently open as file *x*. Examples: RECORD#1,3 selects the third record in the relative file opened as file 1. RECORD#1,3,5 selects the fifth character in the third record. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use RE SHIFT-C 1,3,5.)

SEND (S SHIFT-E) This command has the same effect as OPEN1,8,15: PRINT#1,"*string*": CLOSE1. Example: SEND"I0" initializes the disk drive. SEND "M-R"+CHR$(3) +CHR$(5) reads the byte at location $0503 in the disk drive's memory.

BLOCKS (B SHIFT-L) Displays the number of free blocks remaining on the disk without calling up the entire directory.

PROTECT *"filename"* (PR SHIFT-O) Protects the specified file so that it cannot be scratched. Protected files are denoted on the disk directory with a less-than sign (<). Even a protected file, however, can be erased by reformatting the entire disk. Also, protected program files cannot be read by the TurboDisk utility from the April issue. Attempting to load a protected program with TurboDisk results in a ?FILE NOT FOUND ERROR.

RELEASE *"filename"* (RE SHIFT-L) Unprotects the specified file.

TRANSPOSE *"file1"* WITH *"file2"* (T SHIFT-R) Transposes the positions of two files in the disk directory. **WITH** can be abbreviated **W SHIFT-I.**

Disk Commander is extremely versatile. In addition to letting you imbed the new commands in your programs, it also lets you use them with variables, too. For instance, instead of typing this:

```
DOPEN#1,"filename"
```

you can type this:

```
A=1:A$="filename":DOPEN#A,A$
```

Together with TurboDisk, or just by itself, Disk Commander greatly enhances the power of your Commodore 64.

Commodore 64 Disk Commander

Please refer to the "MLX" article before entering this listing.

```
2Ø49 :Ø11,ØØ8,Ø1Ø,ØØØ,158,Ø5Ø,238
2Ø55 :Ø48,Ø54,Ø49,ØØØ,ØØØ,ØØØ,158
2Ø61 :169,Ø12,133,251,169,16Ø,139
2Ø67 :133,252,162,ØØ2,16Ø,ØØØ,216
2Ø73 :177,251,145,251,2ØØ,2Ø8,233
2Ø79 :249,23Ø,252,2Ø2,2Ø8,244,136
2Ø85 :169,233,133,251,169,161,129
2Ø91 :133,252,169,Ø91,133,253,Ø5Ø
2Ø97 :169,ØØ8,133,254,16Ø,ØØØ,ØØ5
21Ø3 :162,Ø1Ø,177,253,145,251,Ø29
21Ø9 :2ØØ,2Ø8,249,23Ø,252,23Ø,15Ø
2115 :254,2Ø2,2Ø8,242,185,Ø91,225
2121 :Ø18,153,184,ØØ2,2ØØ,192,Ø54
2127 :Ø62,2Ø8,245,Ø32,184,ØØ2,Ø44
2133 :Ø32,Ø68,166,Ø76,116,164,195
2139 :169,158,133,251,169,16Ø,1Ø7
2145 :133,252,169,157,133,253,17Ø
2151 :169,16Ø,133,254,Ø96,Ø32,179
2157 :233,161,Ø76,Ø22,162,169,164
2163 :158,133,251,169,16Ø,133,Ø95
2169 :252,169,157,133,253,169,23Ø
2175 :16Ø,133,254,166,122,16Ø,Ø98
2181 :ØØ4,132,Ø15,189,ØØØ,ØØ2,219
2187 :Ø16,ØØ7,2Ø1,255,24Ø,Ø62,152
2193 :232,2Ø8,216,2Ø1,Ø32,24Ø,25Ø
2199 :Ø55,133,ØØ8,2Ø1,Ø34,24Ø,Ø54
22Ø5 :Ø86,Ø36,Ø15,112,Ø45,2Ø1,14Ø
2211 :Ø63,2Ø8,ØØ4,169,153,2Ø8,2ØØ
2217 :Ø37,2Ø1,Ø48,144,ØØ4,2Ø1,Ø36
2223 :Ø6Ø,144,Ø29,132,113,16Ø,Ø45
2229 :ØØØ,132,Ø11,136,134,122,2Ø4
2235 :2Ø2,2ØØ,232,189,ØØØ,ØØ2,244
2241 :Ø56,241,251,234,24Ø,245,18Ø
2247 :2Ø1,128,2Ø8,Ø48,ØØ5,Ø11,Ø32
2253 :164,113,232,2ØØ,153,251,Ø38
2259 :ØØ1,185,251,ØØ1,24Ø,Ø57,178
2265 :Ø56,233,Ø58,24Ø,ØØ4,2Ø1,241
2271 :Ø73,2Ø8,ØØ2,133,Ø15,Ø56,198
2277 :233,Ø85,2Ø8,131,133,ØØ8,ØØ3
2283 :189,ØØØ,ØØ2,24Ø,223,197,Ø62
2289 :ØØ8,24Ø,219,2ØØ,153,251,Ø32
2295 :ØØ1,232,2Ø8,24Ø,166,122,192
23Ø1 :23Ø,Ø11,2ØØ,177,253,234,Ø78
23Ø7 :Ø16,25Ø,177,251,234,2Ø8,115
2313 :18Ø,Ø76,17Ø,162,189,ØØØ,Ø18
2319 :ØØ2,Ø16,187,153,253,ØØ1,115
2325 :198,123,169,255,133,122,253
2331 :Ø96,165,251,2Ø1,158,2Ø8,Ø82
2337 :235,169,ØØØ,133,251,169,222
2343 :164,133,252,169,255,133,121
2349 :253,169,163,133,254,16Ø,153
2355 :ØØØ,Ø76,Ø76,162,Ø76,Ø96,Ø25
2361 :163,Ø76,1Ø9,163,Ø16,248,Ø64
2367 :2Ø1,255,24Ø,244,Ø36,Ø15,Ø3Ø
2373 :Ø48,24Ø,Ø56,233,127,17Ø,175
2379 :132,Ø73,16Ø,255,224,Ø77,228
2385 :176,Ø22,2Ø2,24Ø,ØØ8,2ØØ,161
2391 :185,158,16Ø,Ø16,25Ø,Ø48,136
2397 :245,2ØØ,185,158,16Ø,Ø48,Ø65
```

```
2403 :214,032,000,168,208,245,198
2409 :056,233,076,170,202,240,058
2415 :008,200,185,000,164,016,172
2421 :250,048,245,200,185,000,021
2427 :164,048,188,032,000,168,211
2433 :208,245,032,115,000,032,249
2439 :026,163,076,015,168,240,055
2445 :062,233,128,144,017,201,158
2451 :035,176,023,010,168,185,232
2457 :013,160,072,185,012,160,243
2463 :072,076,028,168,076,048,115
2469 :168,201,058,240,217,076,101
2475 :086,168,201,075,208,003,144
2481 :076,067,168,176,003,076,231
2487 :086,168,201,095,176,249,134
2493 :233,075,010,168,185,129,221
2499 :163,072,185,128,163,072,210
2505 :076,115,000,096,082,069,127
2511 :065,068,089,072,169,243,145
2517 :133,247,169,166,133,248,029
2523 :104,076,193,002,072,169,067
2529 :239,133,247,169,166,133,032
2535 :248,104,076,193,002,000,086
2541 :000,000,000,000,000,000,237
2547 :165,123,165,180,165,194,211
2553 :165,197,165,247,165,216,124
2559 :166,224,166,232,166,250,179
2565 :166,105,167,198,167,211,251
2571 :167,055,169,070,169,140,013
2577 :169,153,169,081,170,085,076
2583 :168,095,133,034,168,165,018
2589 :091,229,096,170,232,152,231
2595 :240,035,165,890,056,229,082
2601 :034,133,090,176,003,198,163
2607 :091,856,165,088,229,034,198
2613 :133,088,176,008,198,089,233
2619 :144,004,177,090,145,088,195
2625 :136,208,249,177,090,145,046
2631 :088,198,091,198,089,202,169
2637 :208,242,096,010,105,062,032
2643 :176,053,133,034,186,228,125
2649 :034,144,046,096,196,052,145
2655 :144,040,208,004,197,051,227
2661 :144,034,072,162,009,152,162
2667 :072,181,087,202,016,250,147
2673 :032,068,073,082,069,067,248
2679 :084,079,082,217,068,073,210
2685 :083,075,083,212,068,083,217
2691 :065,086,197,068,086,069,190
2697 :082,073,070,217,068,076,211
2703 :079,065,196,083,067,082,203
2709 :065,084,067,200,082,069,204
2715 :078,065,077,197,067,079,206
2721 :080,217,067,079,076,076,244
2727 :069,067,212,072,069,065,209
2733 :068,069,210,068,079,080,235
2739 :069,078,163,065,080,880,202
2745 :069,078,068,163,082,069,202
2751 :067,079,082,068,163,083,221
2757 :069,078,196,066,076,079,249
2763 :867,075,211,080,082,079,029
2769 :084,069,067,212,082,069,024
2775 :076,069,065,083,197,084,021
2781 :082,065,078,083,080,079,176
2787 :083,197,087,073,084,200,183
2793 :000.000,000.080,000.000,233
2799 :000,000,000,107,169,032,035
2805 :121,165,132,011,032,019,213
2811 :166,144,068,160,001,177,199
2817 :095,133,035,165,045,133,095
2823 :034,165,096,133,037,165,125
2829 :095,136,241,095,024,101,193
2835 :045,133,045,133,036,165,064
2841 :046,105,255,133,046,229,071
2847 :096,170,056,165,095,229,074
2853 :045,168,176,003,232,198,091
2859 :037,024,101,034,144,003,130
2865 :198,035,024,177,034,145,150
2871 :036,200,208,249,230,035,245
2877 :230,037,202,208,242,032,244
2883 :089,166,032,051,165,173,231
2889 :008,802,240,136,024,165,128
2895 :045,133,090,101,011,133,080
2901 :088,164,046,132,091,144,238
2907 :001,200,132,089,032,184,217
2913 :163,165,020,164,021,141,003
2919 :254,001,140,255,001,165,151
2925 :049,164,050,133,045,036,074
```

```
2931 :169,001,162,000,160,165,004
2937 :032,189,255,169,001,162,161
2943 :008,160,096,032,186,255,096
2949 :032,192,255,162,001,032,039
2955 :198,255,169,000,133,144,014
2961 :162,005,032,207,255,164,202
2967 :144,208,056,202,208,246,191
2973 :133,251,032,207,255,164,175
2979 :144,208,044,166,251,032,240
2985 :097,168,169,032,032,210,109
2991 :255,032,207,255,164,144,208
2997 :208,027,170,240,006,032,096
3003 :210,255,076,062,165,169,100
3009 :013,032,210,255,032,237,204
3015 :246,240,005,162,003,076,163
3021 :033,165,076,112,168,032,023
3027 :204,255,169,001,076,195,087
3033 :255,162,000,189,118,165,082
3039 :157,004,003,232,224,006,081
3045 :208,245,096,219,002,234,209
3051 :002,240,002,169,008,032,176
3057 :180,255,169,111,032,150,114
3063 :255,032,165,255,032,210,172
3069 :255,201,013,208,246,076,228
3075 :171,255,169,000,032,189,051
3081 :255,162,008,160,001,169,252
3087 :221,141,208,002,169,225,213
3093 :141,209,002,032,199,002,094
3099 :186,169,169,157,003,001,200
3105 :169,167,157,004,001,096,115
3111 :032,147,165,169,089,133,006
3117 :247,169,225,133,248,076,119
3123 :193,002,169,001,044,169,117
3129 :000,133,810,032,147,165,832
3135 :169,111,133,247,169,225,093
3141 :133,248,076,193,002,165,118
3147 :251,208,003,076,086,168,099
3153 :169,008,032,177,255,169,123
3159 :111,032,147,255,160,000,024
3165 :177,252,032,168,255,200,153
3171 :196,251,208,246,076,174,226
3177 :255,032,129,168,032,047,000
3183 :166,169,083,141,080,190,092
3189 :169,058,141,001,190,160,068
3195 :008,177,098,153,002,190,231
3201 :200,196,097,208,246,169,221
3287 :000,133,252,169,190,133,244
3213 :253,200,200,132,251,032,185
3219 :223,165,165,157,016,012,117
3225 :169,013,032,210,255,076,140
3231 :124,165,165,157,048,001,051
3237 :096,160,000,185,113,166,117
3243 :032,210,255,280,192,014,050
3249 :208,245,032,204,255,032,129
3255 :207,255,201,089,208,025,144
3261 :032,207,255,201,013,240,113
3267 :225,201,069,288,014,032,176
3273 :207,255,201,083,208,007,138
3279 :032,207,255,201,013,240,131
3285 :207,201,013,240,006,032,144
3291 :207,255,208,247,096,104,056
3297 :104,096,065,082,069,032,161
3303 :089,079,085,032,083,085,172
3309 :082,069,063,032,032,129,132
3315 :165,160,000,177,098,153,231
3321 :000,191,200,196,097,144,053
3327 :246,132,250,160,000,177,196
3333 :122,201,164,240,003,076,043
3339 :086,168,032,115,000,032,188
3345 :129,168,160,000,177,098,237
3351 :153,002,190,200,196,097,093
3357 :144,246,169,061,153,002,036
3363 :190,200,200,200,132,252,185
3369 :169,190,133,253,160,000,178
3375 :185,000,191,145,252,200,252
3381 :196,250,144,246,152,024,841
3387 :101,252,133,251,169,000,197
3393 :133,252,169,058,141,001,051
3399 :190,076,223,165,169,082,208
3405 :141,000,190,076,127,166,009
3411 :169,067,141,000,190,076,214
3417 :127,166,169,008,032,177,000
3423 :255,169,111,032,147,255,040
3429 :169,086,032,168,255,076,119
3435 :174,255,032,129,168,032,129
3441 :047,166,169,078,141,000,202
3447 :190,169,058,141,001,190,180
3453 :160,000,177,098,153,002,203
```

```
3459 :190,200,196,097,208,246,244
3465 :169,000,133,252,169,190,026
3471 :133,253,200,200,132,251,032
3477 :032,223,165,165,157,016,139
3483 :008,169,013,032,210,255,074
3489 :076,124,165,096,032,152,038
3495 :168,160,000,177,122,201,227
3501 :044,240,003,076,086,168,022
3507 :134,184,032,165,168,165,003
3513 :097,133,183,169,000,133,132
3519 :187,169,190,133,188,160,194
3525 :000,177,098,145,187,200,236
3531 :196,183,208,247,169,044,226
3537 :145,187,200,169,083,145,114
3543 :187,200,132,183,096,032,021
3549 :051,167,032,121,000,201,025
3555 :044,208,076,032,115,000,190
3561 :201,087,208,040,169,044,214
3567 :145,187,200,169,087,145,148
3573 :187,200,132,183,032,115,070
3579 :000,169,008,133,186,160,139
3585 :097,200,152,166,152,202,202
3591 :048,007,221,045,006,240,062
3597 :244,208,246,132,185,076,080
3603 :192,255,201,876,208,194,121
3609 :032,171,168,169,076,164,037
3615 :183,136,145,187,200,169,027
3621 :044,145,187,200,138,145,128
3627 :187,208,132,183,076,138,191
3633 :167,198,183,198,183,076,030
3639 :138,167,032,851,167,169,011
3645 :044,145,187,200,169,065,103
3651 :076,184,167,032,152,168,078
3657 :032,121,000,201,044,240,199
3663 :003,076,886,168,134,251,029
3669 :032,177,168,169,001,133,253
3675 :252,032,121,000,201,044,229
3681 :208,005,032,171,168,134,047
3687 :252,076,000,169,000,000,088
3693 :008,000,000,000,000,072,181
3699 :169,071,141,208,002,169,107
3705 :171,141,209,002,104,076,056
3711 :199,002,072,169,174,133,108
3717 :247,169,167,133,248,104,177
3723 :076,193,002,186,169,233,230
3729 :157,003,001,169,167,157,031
3735 :004,001,169,000,072,169,054
3741 :114,072,076,225,002,104,238
3747 :104,169,167,072,169,233,053
3753 :072,169,165,133,247,169,108
3759 :169,133,248,076,193,002,228
3765 :104,104,169,167,072,169,198
3771 :233,072,169,018,133,247,035
3777 :169,168,133,248,076,193,156
3783 :002,169,008,133,247,169,159
3789 :175,133,248,076,193,082,008
3795 :072,169,205,141,208,002,240
3801 :169,189,141,209,002,104,007
3807 :076,199,002,169,052,133,086
3813 :247,169,168,133,248,169,083
3819 :001,032,195,255,056,076,082
3825 :193,002,169,158,141,208,088
3831 :002,169,173,141,209,002,175
3837 :032,199,002,072,169,143,102
3843 :141,208,002,104,076,221,243
3849 :168,169,158,141,208,002,087
3855 :169,183,141,209,002,076,027
3861 :199,002,032,115,000,076,189
3867 :129,168,032,115,000,076,035
3873 :152,168,032,115,000,169,157
3879 :138,141,208,002,169,173,102
3885 :141,209,002,032,199,002,118
3891 :072,169,247,141,208,002,122
3897 :169,183,141,209,002,104,097
3903 :076,199,002,169,055,133,185
3909 :247,169,164,133,248,162,168
3915 :803,076,193,002,032,199,068
3921 :002,072,169,163,141,208,068
3927 :002,169,182,141,209,002,024
3933 :104,832,199,002,133,097,148
3939 :134,098,132,099,096,000,146
3945 :000,000,016,000,000,000,121
3951 :000,255,000,165,251,166,180
3957 :152,202,016,003,076,208,006
3963 :168,221,089,002,208,245,032
3969 :189,109,002,141,001,190,249
3975 :169,080,141,000,190,165,112
3981 :020,141,002,190,165,021,168
```

```
3987 :141,003,190,165,252,141,015
3993 :004,190,169,005,133,251,137
3999 :169,000,133,252,169,190,048
4005 :133,253,076,223,165,032,023
4011 :129,168,162,003,181,096,142
4017 :149,250,202,208,249,076,031
4023 :223,165,169,073,141,000,186
4029 :190,169,001,032,043,169,025
4035 :162,000,189,135,169,157,239
4041 :000,190,232,224,006,208,037
4047 :245,169,006,032,043,169,103
4053 :169,008,032,180,255,169,002
4059 :111,032,150,255,169,013,181
4065 :032,210,255,032,165,255,150
4071 :170,032,165,255,032,165,026
4077 :255,032,097,168,032,171,224
4083 :255,169,013,076,210,255,197
4089 :077,045,082,250,002,003,196
4095 :169,009,141,048,170,169,193
4101 :064,141,049,170,076,164,157
4107 :169,169,041,141,048,170,237
4113 :169,191,141,049,170,032,001
4119 :061,170,169,002,162,008,083
4125 :160,170,032,189,255,162,229
4131 :008,160,002,032,186,255,166
4137 :032,192,255,162,000,189,103
4143 :010,170,157,000,190,232,038
4149 :224,008,208,245,134,251,099
4155 :032,045,169,162,002,032,245
4161 :201,255,162,000,189,018,122
4167 :170,032,210,255,232,224,170
4173 :043,208,245,032,204,255,040
4179 :032,129,168,160,000,177,237
4185 :098,153,003,190,200,196,161
4191 :097,208,246,152,024,105,159
4197 :003,133,251,169,085,141,115
4203 :000,190,169,051,141,001,147
4209 :190,169,058,141,002,190,095
4215 :076,074,170,035,050,066,078
4221 :045,080,032,050,032,048,156
4227 :013,032,238,193,032,152,023
4233 :195,032,032,195,032,202,057
4239 :195,169,000,133,134,032,038
4245 :157,196,048,016,032,183,013
4251 :221,144,014,160,000,177,103
4257 :148,009,064,145,148,076,239
4263 :187,200,076,225,202,076,109
4269 :087,217,032,231,255,169,140
4275 :073,141,000,190,169,001,241
4281 :076,043,169,032,045,169,207
4287 :169,002,076,195,255,032,152
4293 :061,170,169,002,162,008,001
4299 :160,170,032,189,255,162,147
4305 :008,160,002,032,186,255,084
4311 :032,192,255,162,000,189,021
4317 :010,170,157,000,190,232,212
4323 :224,008,208,245,134,251,017
4329 :032,045,169,162,002,032,163
4335 :201,255,162,000,189,041,063
4341 :171,032,210,255,232,224,089
4347 :159,208,245,032,204,255,074
4353 :032,129,168,160,000,177,155
4359 :098,153,003,190,200,196,079
4365 :097,208,246,152,024,105,077
4371 :003,133,251,169,085,141,033
4377 :000,190,169,052,141,001,066
4383 :190,169,058,141,002,190,013
4389 :032,251,170,032,121,000,131
4395 :201,222,240,003,076,086,103
4401 :168,032,115,000,032,129,013
4407 :168,160,000,177,098,153,043
4413 :003,190,200,196,097,208,187
4419 :246,152,024,105,003,133,218
4425 :251,169,085,141,000,190,141
4431 :169,053,141,001,190,169,034
4437 :058,141,002,190,032,251,247
4443 :170,169,002,133,251,169,217
4449 :085,141,000,190,169,051,221
4455 :141,001,190,076,074,170,243
4461 :032,045,169,169,008,032,052
4467 :180,255,169,111,032,150,244
4473 :255,032,165,255,201,048,053
4479 :208,005,032,165,255,201,225
4485 :048,008,032,171,255,040,175
4491 :208,001,096,162,004,169,011
4497 :055,133,247,169,164,133,022
4503 :248,076,193,002,076,069,047
4509 :005,160,000,044,160,033,047
4515 :140,176,005,032,238,193,179
4521 :032,152,195,032,032,195,039
4527 :032,202,195,032,157,196,221
4533 :016,003,076,225,202,165,100
4539 :148,174,176,005,157,177,000
4545 :005,165,149,010,168,185,107
4551 :000,000,157,178,005,185,212
4557 :001,000,157,179,005,160,195
4563 :000,177,148,157,180,005,110
4569 :232,200,192,030,208,245,044
4575 :096,173,178,005,133,006,046
4581 :173,179,005,133,007,169,127
4587 :128,133,000,162,000,032,178
4593 :153,213,160,036,174,177,130
4599 :005,185,177,005,157,000,008
4605 :003,232,200,192,066,208,130
4611 :244,169,144,133,000,162,087
4617 :000,032,153,213,173,211,023
4623 :005,133,006,173,212,005,037
4629 :133,007,169,128,133,000,079
4635 :162,000,032,153,213,160,235
4641 :003,174,210,005,185,177,019
4647 :005,157,000,003,232,200,124
4653 :192,033,208,244,169,144,011
4659 :133,000,162,000,076,153,063
4665 :213,239,255,255,255,255,249
4671 :255,255,255,255,255,255,057
4677 :255,255,255,255,255,255,063
4683 :255,255,255,255,255,255,069
4689 :255,000,000,000,000,000,080
4695 :000,000,000,000,032,210,073
4701 :002,032,104,165,076,225,185
4707 :002,032,225,002,108,247,203
4713 :000,072,165,001,009,001,097
4719 :133,001,104,032,205,189,007
4725 :072,165,001,041,254,133,015
4731 :001,104,096,032,210,002,056
4737 :032,000,162,072,165,001,049
4743 :009,001,133,001,104,096,223
4749 :032,210,002,076,203,162,058
4755 :032,210,002,076,017,163,135
```

©

Apple Fractals

Paul W. Carlsan

Fractals are receiving a great deal of attention in mathematics and computer graphics these days. They're being used for everything from simulating random plant growth to generating realistic planetary landscapes for science-fiction films. This article introduces the fascinating world of fractals with three programs that demonstrate a particular type of fractal that can be plotted on a personal computer.

The word *fractal* was coined by Benoit Mandelbrot, a pioneer in their study, to denote curves or surfaces having *fractional dimension.* The concept of fractional dimension can be illustrated as follows: A straight curve (a line) is one-dimensional, having only length. However, if the curve is infinitely long and curves about in such a manner as to completely fill an area of the plane containing it, the curve could be considered two-dimensional. A curve partially filling an area would have a fractional dimension between one and two.

Many types of fractals are *self-similar,* which means that all portions of the fractal resemble each other. Self-similarity occurs whenever the whole is an expansion of some basic building block. In the language of fractals, this basic building block is called the *generator*. The generator in the accompanying programs consists of a number of connected line segments. The curves that the programs plot are the result of starting with the generator and then repeatedly replacing each line segment with the whole generator according to a defined rule. Theoretically, these replacement cycles would continue indefinitely. In practice, the screen resolution limits the number of cycles.

The programs illustrate two types of fractal curves. The curves generated by Program 1 and Program 2 are *self-contacting,* while the curve generated by Program 3 is *self-avoiding.* A self-contacting curve touches itself but does not cross itself. A self-avoiding curve never actually touches itself although it may appear to because of the limited screen resolution.

The Dragon Sweep

Program 1 plots what Mandelbrot refers to as a "dragon sweep." It demonstrates in a step-by-step fashion how a fractal curve is filled. The generator consists of two-line segments of equal length forming a right angle. During each replacement cycle, the generator is substituted for each segment on alternating sides of the segments, that is, to the left of the first segment, to the right of the second segment, and so on. Figure 1 shows the first few cycles of substitution. The program is written in BASIC so the plotting is slow enough to let you observe the development of the curve.

The program prompts you to enter an even number of cycles (for reasons of efficiency and screen resolution, only even numbers of cycles are plotted). When a plot is complete, pressing any key clears the screen and returns you to the prompt. I recommend starting with two cycles, then four, six, etc. It takes fourteen cycles to completely fill in the "dragon," but since this requires almost two hours, you will probably want to quit after about ten cycles. You can see the complete dragon by running Program 2, which always plots the dragon first in less than 30 seconds.

Since it's not at all obvious how the program works, here's a brief explanation. NC is the number of cycles; C is the cycle number; SN is an array of segment numbers indexed by cycle number; L is the segment length; D is the segment direction, numbered clockwise from the positive x direction; and X and Y are the high-resolution screen coordinates.

Lines 100–140	Get number of cycles from user.
Line 150	Computes segment length.
Line 160	Sets starting coordinates.
Line 170	Sets segment numbers for all cycles to the first segment.
Lines 180–220	Find the direction of the segment in the last cycle by rotating the segment in each cycle that will contain the segment in the last cycle.
Lines 230–260	Increase or decrease X or Y by the segment length, depending on the segment direction.

Lines 270-290	Plot the segment and update the current segment number for each cycle.
Lines 300-320	If the segment number for cycle zero is still zero, do the next segment; otherwise, we're done.

Eight Thousand Dragons

Program 2 plots more than 8,000 different dragons. It does this by randomly determining on which side of the first segment the generator will be substituted for all cycles after the first cycle. The generator is always substituted to the left of the first segment in the first cycle to avoid plotting off the screen. Other than the randomization, this program uses the same logic as Program 1. The main part of this program is written in machine language to reduce the time required to plot a completely filled-in dragon from about two hours to less than half a minute.

All the dragons are plotted after fourteen cycles of substitution. All have exactly the same area, which equals half of the square of the distance between the first and last points plotted. All the dragons begin and end at the same points.

When a plot is complete, press the space bar to plot another dragon, or press the Q key to quit.

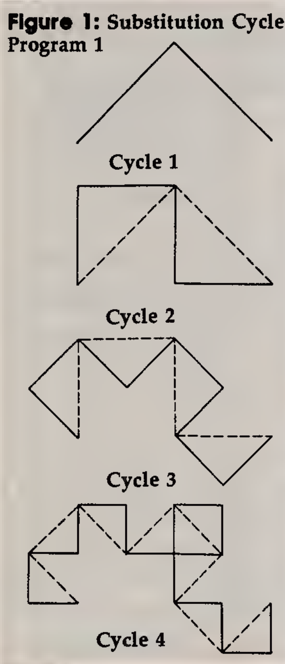
Snowflakes

Program 3 plots what Mandelbrot refers to as a "snowflake sweep." The generator, shown in Figure 2, was discovered by Mandelbrot. The segments are numbered zero through six, starting at the right. The program is basically the same as Program 1. The variables NC, C, SN, D, X, and Y represent the same values except that the direction D is numbered counterclockwise from the negative x direction. For each segment, the accompanying table gives the value of RD (relative direction), LN (length factor), and SD (flags indicating which side of the segment the generator is to be placed).

Line 20	Reads values of SD and RD. Compute LN values.
Lines 30-50	Compute delta x and delta y factors for each direction.
Lines 60-100	Get number of cycles from user.
Line 120	Sets starting coordinates.
Line 130	Sets the segment numbers for all cycles to the first segment.
Lines 140-170	Find the direction of the segment in the last cycle.
Lines 180-190	Compute the coordinates of the end of the segment, plot the segment, and update the segment numbers for each cycle.
Lines 200-220	Same as lines 300–320 in Program 1.

Like Program 1, pressing any key when a plot is complete clears the screen and brings another prompt.

Figure 1: Substitution Cycles, Program 1



Cycle 1

Cycle 2

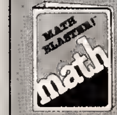
Cycle 3

Cycle 4

Experiment!

I hope these programs encourage you to look further into the fascinating world of fractals. Don't be afraid to experiment with the programs—try modifying the shape of the generator in Program 3, for example. Better yet, design your own generator.

These programs just begin to explore the possibilities of fractal computer graphics. There is another whole class of fractals, those generated by functions of complex variables. And then there are three-dimensional fractals. And then . . . .



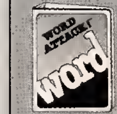
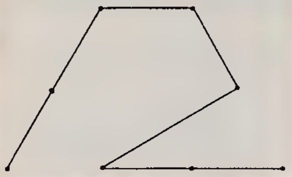


Figure 2: Generator, Program 3



Values For Program 3

Segment Number SN	Relative Direction RD	Length Factor LN	Side Flag SD
0	0	1/3	0
1	0	1/3	1
2	7	$\sqrt{1/3}$	1
3	10	1/3	0
4	0	1/3	0
5	2	1/3	0
6	2	1/3	1

Program 1: The Dragon Sweep

```
1E 10 REM  PRDGRAM 1
6A 20 REM
78 30 REM  THIS PROGRAM PLOTS A
       FRACTAL "DRAGDN SWEEP"
DØ 40 REM  FDR AN EVEN NUMBER DF
        CYCLES (14 MAX).
6D 50 REM
9D 90 DIM SN(14)
54 100 TEXT : HDME
F1 110 PRINT "ENTER AN EVEN ND.
        DF CYCLES (2 TD 14)"
9Ø 120 INPUT "                OR ENTER
         A ZERO TO QUIT: ";NC
A7 130 IF NC = Ø THEN END
E4 140 IF INT (NC / 2) * 2 < > N
        C DR NC < 2 OR NC > 14 TH
        EN 100
1D 150 L = 128: FOR C = 2 TD NC
        STEP 2:L = L / 2: NEXT
E8 160 X = 77:Y = 128: HGR2 : HC
        DLDR= 3: HPLOT X,Y
81 170 FDR C = Ø TD NC:SN(C) = Ø
        : NEXT
43 180 D = Ø: FDR C = 1 TD NC: I
        F SN(C - 1) = SN(C) THEN
        D = D - 1: GDTD 200
46 190 D = O + 1
ED 200 IF D = - 1 THEN D = 7
1C 210 IF O = 8 THEN D = Ø
FD 220 NEXT
9D 230 IF D = Ø THEN X = X + L:
        GOTO 270
FØ 240 IF O = 2 THEN Y = Y + L:
        GOTO 270
A4 250 IF O = 4 THEN X = X - L:
        GOTO 270
9A 260 Y = Y - L
35 270 HPLOT TO X,Y:SN(NC) = SN(
        NC) + 1
1Ø 280 FOR C = NC TO 1 STEP - 1:
         IF SN(C) < > 2 THEN 300
9F 290 SN(C) = Ø:SN(C - 1) = SN(
        C - 1) + 1: NEXT
8A 300 IF SN(Ø) = Ø THEN 180
D6 310 GET A$: IF A$ = "" THEN 3
        10
9Ø 320 GOTO 100
```

Program 2: Eight Thousand Dragons

```
2E 10 REM  PRDGRAM 2
6A 20 REM
68 30 REM
92 40 REM  THIS PROGRAM PLDTS RA
       NDDM FRACTAL "DRAGDN SWEEP
       S."
7C 50 REM  THE "STANDARD" DRAGDN
        IS ALWAYS PLDTTED FIRST.
6E 60 REM
5F 70 REM  WHEN A PLOT IS COMPLE
       TE, PRESS THE SPACE 8AR
D1 80 REM  TO PLOT ANDTHER DRAGD
       N, OR PRESS THE "Q" KEY
97 90 REM  TD EXIT THE PROGRAM.
82 100 REM
8B 130 REM
68 140 HIMEM: 16383
D8 150 FOR N = 24612 TD 24912: R
        EAD I: PDKE N,I: NEXT
9F 160 FDR N = 24591 TD 24605: P
        DKE N,Ø: NEXT : GOTO 180
17 170 FDR N = 24593 TD 24605: P
        DKE N, INT ( RND (1) * 2)
        : NEXT
24 180 HGR2 : HCOLOR= 3: CALL 24
        619
85 190 GET A$: IF A$ = " " THEN
        170
D8 200 IF A$ < > "Q" THEN 190
FF 210 TEXT : END
FØ 220 DATA 1,2,4,8,16,32,64,169
64 230 DATA Ø,141,16,96,160,14,1
        53,Ø
1C 240 DATA 96,136,192,255,208,2
        48,141,32
AF 250 DATA 96,162,77,142,31,96,
        160,128
22 260 DATA 140,33,96,32,248,96,
        169,Ø
A5 270 DATA 141,30,96,162,Ø,160,
        1,185
D8 280 DATA 15,96,208,20,238,30,
        96,189
26 290 DATA Ø,96,217,Ø,96,208,26
        ,206
28 300 DATA 30,96,206,30,96,76,1
        25,96
A8 310 DATA 206,30,96,189,Ø,96,2
        17,Ø
26 320 DATA 96,208,6,238,30,96,2
        38,30
B5 330 DATA 96,173,30,96,16,5,16
        9,7
AF 340 DATA 141,30,96,201,8,208,
        5,169
16 350 DATA Ø,141,30,96,232,200,
        224,14
D8 360 DATA 208,189,170,208,20,1
        73,31,96
Ø7 370 DATA 24,105,1,141,31,96,1
        73,32
4D 380 DATA 96,105,Ø,141,32,96,7
        6,210
7A 390 DATA 96,224,2,208,6,238,3
        3,96
44 400 DATA 76,210,96,224,4,208,
        20,173
ØC 410 DATA 31,96,56,233,1,141,3
        1,96
53 420 DATA 173,32,96,233,Ø,141,
        32,96
E1 430 DATA 76,210,96,206,33,96,
        32,248
15 440 DATA 96,238,14,96,160,14,
        162,13
68 450 DATA 185,Ø,96,201,2,208,1
        2,169
B4 460 DATA Ø,153,Ø,96,254,Ø,96,
        202
CF 470 DATA 136,208,237,173,Ø,96
        ,208,3
E1 480 DATA 76,74,96,96,173,33,9
        6,10
D1 490 DATA 10,41,28,9,64,133,27
        ,173
28 500 DATA 33,96,74,74,74,74,41
        ,3
FF 510 DATA 5,27,133,27,173,33,9
        6,41
45 520 DATA 192,72,106,133,26,10
        4,74,74
1F 530 DATA 74,5,26,133,26,173,3
        1,96
8F 540 DATA 141,34,96,173,32,96,
        141,35
66 550 DATA 96,56,160,255,200,17
        3,34,96
ØC 560 DATA 233,7,141,34,96,173,
        35,96
35 570 DATA 233,Ø,141,35,96,16,2
        37,173
FC 580 DATA 34,96,105,7,170,189,
        36,96
71 590 DATA 17,26,145,26,96
```

Program 3: The Snowflake Sweep

```
3E 10 REM  PRDGRAM 3
6A 20 REM
8Ø 30 REM  THIS PROGRAM PLDTS A
       FRACTAL "SNDWFLAKE SWEEP"
6C 40 REM
9C 50 DIM DX(11),DY(11):M = 7 /
       6
1C 60 FDR N = Ø TO 6: READ SD(N)
       ,RD(N):LN(N) = 1 / 3: NEXT
        :LN(2) = SQR (LN(1))
F1 70 A = Ø: FOR D = 6 TD 11:DX(
       D) = CDS (A):OY(D) = SIN (
       A)
BC 80 A = A + Ø.52359879: NEXT
E8 90 FDR D = Ø TD 5:DX(D) = - D
       X(O + 6):OY(D) = - OY(D +
       6): NEXT
54 100 TEXT : HOME
85 110 PRINT "ENTER NUMBER OF CY
        CLES ( 1 - 4 )"
9Ø 120 INPUT "            OR ENTER
        A ZERO TD QUIT: ";NC
A7 130 IF NC = Ø THEN END
1A 140 IF NC > 4 THEN 100
9D 150 HGR2 : HCDLDR= 3
BE 160 X = 235:Y = 142:TL = 162:
         HPLDT X,Y
81 170 FDR C = Ø TD NC:SN(C) = Ø
        : NEXT
Ø4 180 D = Ø:L = TL:NS = Ø: FDR
        C = 1 TD NC:I = SN(C):L =
         L * LN(I):J = SN(C - 1):
        NS = NS + SD(J):K = INT (
        NS / 2): IF K * 2 < > NS
        THEN O = O + 12 - RD(I):
        GDTO 200
61 190 D = O + RO(I)
92 200 IF O > 11 THEN O = O - 12
F8 210 NEXT
7Ø 220 X = X + M * L * OX(D):Y =
         Y - L * OY(O): HPLDT TO
        X,Y:SN(NC) = SN(NC) + 1:
        FDR C = NC TD 1 STEP - 1:
         IF SN(C) < > 7 THEN 240
93 230 SN(C) = Ø:SN(C - 1) = SN(
        C - 1) + 1: NEXT
C1 240 IF SN(Ø) = Ø THEN 180
4E 250 GET A$: IF A$ = "" THEN 2
        50
97 260 GOTO 100
41 270 DATA Ø,Ø,1,Ø,1,7,Ø,10,Ø,Ø
        ,Ø,2,1,2
```

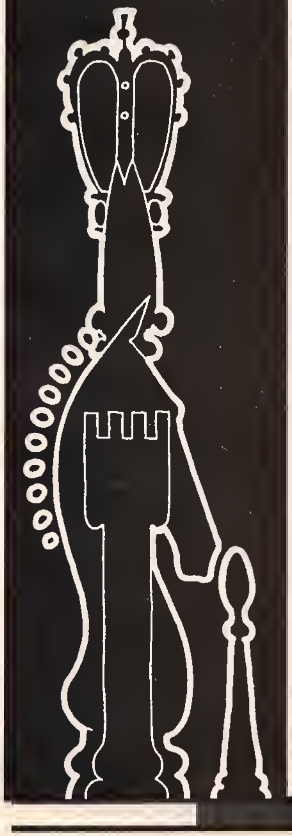
©

For IBM PC & PCjr

Chess

John Krouse, Assistant Technical Editor

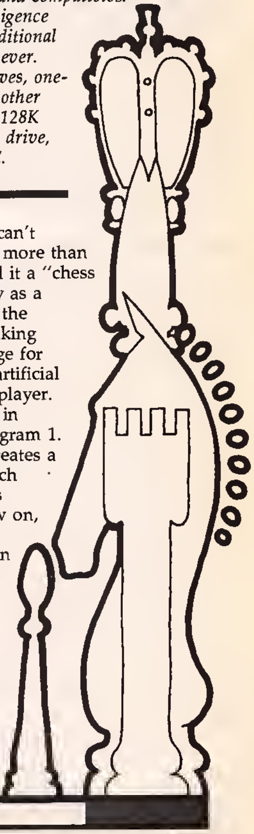
In the December 1984 issue, COMPUTE! *published "Chess" for the Commodore 64, VIC-20, Atari, and Apple computers. This month, by popular demand, we present an all-new version for the IBM PC, PCjr, and compatibles. Like the original Chess, the IBM version has intelligence routines written entirely in machine language. Additional features make it our most powerful chess program ever. It has multiple skill levels, checking for illegal moves, one- and two-player modes, reverse moving, and many other features. The program requires a PC with at least 128K RAM, color/graphics adapter, BASICA, and a disk drive, or an Enhanced Model PCjr with Cartridge BASIC.*

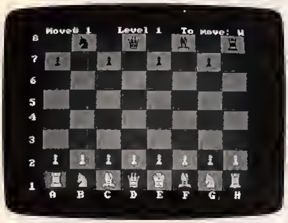


A computer chess game is great for those who can't always find a human opponent. But "Chess" is more than just a substitute for a live player. You might call it a "chess processor." It processes chess positions as easily as a word processor manipulates text. It contains all the features a chess player could ever want. Its thinking routines are written entirely in machine language for greater speed, and they use basic principles of artificial intelligence to simulate an actual human chess player.

Chess consists of two programs. First, type in and save each program. Then load and run Program 1. You'll have to wait about 15 seconds while it creates a BLOAD file on the disk called CHESS.BLD which contains the machine language. Once this file is created, Program 1 is no longer used. From now on, to play Chess, simply load and run Program 2.

After running Chess, you'll see a title screen for a few seconds while the computer prepares itself. Then the board is displayed with the pieces in their starting positions. You're in command of the white pieces versus the computer's black pieces on skill level 1, the easiest level. You should see a frame around the square in the lower-left corner of the board. This is the cursor which takes the place of your hand for moving and capturing pieces.





Move# 1 Level 1 To Move: W

"Chess" for the IBM PC and PCjr is COMPUTE!'s *most powerful chess program to date.*

Use the cursor keys to move the frame cursor atop the piece you wish to move. Press and release the Enter key. Now move the cursor to the square on which you want to place the piece and hit Enter again. Your piece moves to the new square, and the computer responds instantly with a countermove.

Sorry, No Cheating

One of the most valuable features of IBM Chess is that it checks for illegal moves. If you try to make an illegal move, the computer buzzes and keeps your piece on its square. This feature is not perfect, however. It won't catch illegal moves involving castling or *en passant* captures. But it will catch 99 percent of all illegal moves, including those that put your king in check, as well as the more obvious ones such as moving a pawn backwards. If the computer accepts your move, it's probably legal, but not necessarily so. If the computer rejects your move, however, you can be sure that it is illegal.

If you're a beginner at chess, you'll find the move-checking feature especially valuable. Just by trying various moves and noting which ones the computer accepts, you can get a good idea of the way each piece can move.

Information about the current game is displayed at the top of the screen. *Move#* indicates the number of the move currently being made, counting from the start of the game. In chess, a move by both sides is considered one move. So, the move number is changed only after both sides have moved.

To Move indicates which side has the move. W means it is white's turn, and B means it is black's.

Normally after you move, the computer automatically makes the next move. This can be turned off by pressing the T key to switch to two-player mode. Now you can play against another person with the computer acting as referee to check for illegal moves. To switch back to one-player mode, press T again.

You can also let the computer make moves for you by pressing the M key. The side that the computer plays depends on whose turn it is. By repeatedly pressing M, you can watch the computer play itself.

Five Skill Levels

One of the advantages of a computer opponent over a human is that you can tell the computer exactly how hard you want it to try to beat you, and it obediently plays at that level of difficulty. This is important because it's no fun if you always lose or always win effortlessly.

Level shows the current skill level from 1 to 5. You can change the level at any time by pressing keys 1–5. The difference between levels is the number of moves ahead that the computer looks. On level 1, for example, it looks ahead one full move or two half-moves (its move and your reply). Each succeeding level looks ahead one more half-move than the previous level.

Alas, the smarter play on the higher levels doesn't come without a price. The further ahead the computer looks, the more moves it must examine and, hence, the longer it thinks. Here's a rundown of the five levels:

Level 1: Beginner. Thinking time: one second. Look-ahead: two half-moves. Fast but dumb.

Level 2: Intermediate. Thinking time: five seconds. Look-ahead: three half-moves. Provides a reasonable challenge for impatient players.

Level 3: Tournament. Thinking time: two minutes. Look-ahead: four half-moves. Since the usual time limit for tournament play is 40 moves in two hours, an average of three minutes per move, this level is best suited for serious players.

Level 4: Mate in two. Thinking time: 20 minutes. Look-ahead: five half-moves. Capable of solving most mate-in-two problems.

Level 5: Postal chess. Thinking time: two hours. Look-ahead: six half-moves. Simulates chess by mail where there is no time limit. Can avoid checkmate in two moves.

These thinking times are averages. The actual thinking time varies greatly depending on the position. For example, level 5 takes only five seconds with just two kings on the board. Also, these times are for the PC only. Since the PCjr runs at about two-thirds the speed of the PC, the thinking times for the PCjr are greater than the values shown above.

A Spectacular Blunder

It happens to everyone. It's inevitable. You've played for an hour, somehow managing to maneuver into a superior position in what you consider to be the best game of your life, only to throw it all away in a single, spectacular blunder.

Don't panic. You can take back the last half-move by pressing the B key. If you're in one-player mode, you need to press B again to take back your move and the computer's reply. In fact, you can press B repeatedly to take back several moves until you reach the starting position. This is possible because the computer records every move made in the game.

Another use for this feature is to allow the computer to suggest a move for you. If you don't have a good idea of where to move next, press M and the computer will move for you. If you like that move, press M again to continue with the computer's next move. But if you think you've found a better move, press B to take back the suggested move and make your own move.

Pressing the F key does the opposite of B. It moves forward through the move list up to the most advanced position. Note that every time a new move is made, the resulting position becomes the most advanced. So if you use B to backtrack to a previous position, and then make a new move, all subsequent stored moves are erased because they are no longer relevant.

If you have a printer, you can print the move list by pressing the P key. The list appears in three columns: the move numbers, white's moves, and black's moves. Each move is indicated by the square the piece moved from followed by the square it moved to. Each square is specified by its coordinates according to the numbers along the left side of the board and letters along the bottom.

You can also dump the screen image to the printer to get a hardcopy of a particularly interesting position. Before loading BASIC from DOS, type GRAPHICS with the DOS master disk in the drive. Then run Chess and press Shift-PrtSc (Fn-PrtSc on the PCjr) whenever you want to print the position.

Checkmate

The computer thinks by analyzing thousands of possible moves and countermoves and choosing what it considers to be the best move based on the relative value of the pieces. Most positions don't have just one best move but several which are equally good, in which case the computer chooses among them at random. This random factor insures that every game will be different, and makes for varied and interesting play.

The computer announces checkmate when it occurs. However, there are a few quirks in the way the computer evaluates a checkmate. On levels 3–5, it announces checkmate prematurely. When this happens, the computer has determined that it's impossible to avoid checkmate on the *next* move or two—assuming both sides make the best moves.

Also, the computer doesn't know the subtle difference between checkmate and stalemate. Consequently, when a game is stalemated, the computer announces checkmate even though the game is a draw. Since the computer tries as hard as it can to checkmate its opponent, it also tries to achieve stalemate, possibly forcing a draw when it could have won. Fortunately, this rarely happens, because a stalemate requires unusual circumstances, such as when one side has only the king remaining.

You can start a new game at any time by pressing the N key. This sets up the pieces in the starting position with white on the bottom. If you want to play the black pieces, you can press the I key to invert the board, so you still play from the bottom. As with the N command, the board is reset to the starting position. However, the N and I commands retain the move list from the previous game. This allows you to replay the game using the F command. When replaying a game, be sure to reset the board by pressing I if the game was played in the inverted mode, or N if normal mode was used.

Set Up Any Position

You don't have to begin a game from the starting position. You can set up any position and begin playing from that point. If you want, you can first clear the board by pressing the C key. To add a piece or change a piece to a different one, move the cursor to the appropriate square, hold down either Shift or Ctrl, and press P, N, B, R, Q, or K for pawn, knight, bishop, rook, queen, or king, respectively. Holding down Shift adds one of the lower player's pieces, and Ctrl adds one of the upper player's pieces. (Just remember that Ctrl is above Shift on the keyboard.) A piece can be removed from the board by pressing the space bar. Note that these changes are not stored in the move list.

These commands allow you to experiment with hypothetical or downright ridiculous positions. The position doesn't even have to be legal. Live out your fantasy by giving yourself ten queens versus the computer's lone king. Or invent your own type of chess by giving each side two kings, for example (although in this case the computer might get confused trying to determine a checkmate).

You can also set up a problem for the computer to solve, such as the mate-in-two problems published in many newspapers. To solve a mate-in-two problem, press C to clear the board, set up the position, press 4 to select level 4, and press M to start the computer thinking. After several minutes of deep thought, the computer will make a move (the solution) and announce checkmate. The only mate-in-two problems that the computer cannot solve are those which involve castling, *en passant* captures, or pawn promotion.

Special Moves

The computer never castles or captures *en passant* because, due to their complexity, these moves are not included in its thinking routine. But *you* can make these special moves. To castle, move the king *two* squares to the left or right. The rook moves automatically. To capture *en passant,* move your pawn diagonally to the proper square. The opponent's pawn is removed automatically. Remember, the computer doesn't check for illegal moves involving castling or *en passant* captures, so if you're a beginner, you should familiarize yourself with the rules on these special moves.

When a pawn reaches the opposite side of the board, it's automatically promoted to a queen. In the rare event that you would rather promote to a knight, bishop, or rook, you can easily make the change by positioning the cursor over the new queen and pressing N, B, or R with Shift or Ctrl. Note, however, that underpromotions are not stored in the move list.

Saving A Game

If you want to stop the present game and continue later, you can save the game on disk (in drive A) by pressing the S key. You'll see the prompt *Save:*. Type in a filename for your game and press Enter. The filename can be up to eight characters long. Don't type an extender; .CHS is added automatically. If a file on the disk already has the same name, it will be replaced.

To load a previously saved game, press the L key. Answer the *Load:* prompt with the filename and press Enter. (Don't type the .CHS extender.) The L command restores the game exactly as it was when it was saved. Not only the position is restored, but also the move list and even the position of the cursor.

If the computer is unable to save or load a game, an error number is displayed. See Appendix A of the *BASIC Reference Manual* for a description of the error.

Besides allowing you to continue a game at a later time, the S and L commands can be used to create a library of your best games. To do this, press N or I just before saving. The game will come up in the starting position when loaded and can be replayed using the F command.

IBM Chess Commands

B: Move backward
C: Clear board
F: Move forward
I: New game (inverted)
L: Load game
M: Computer's move
N: New game
P: Print move list
S: Save game
T: Two players
1–5: Level
Cursor Keys: Move cursor
Enter: Your move
Space Bar: Remove piece
Shift-P: Lower player's pawn
Shift-N: Lower player's knight
Shift-B: Lower player's bishop
Shift-R: Lower player's rook
Shift-Q: Lower player's queen
Shift-K: Lower player's king
Ctrl-P: Upper player's pawn
Ctrl-N: Upper player's knight
Ctrl-B: Upper player's bishop
Ctrl-R: Upper player's rook
Ctrl-Q: Upper player's queen
Ctrl-K: Upper player's king

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: IBM Chess (Machine Language)

```
IG 10 OEF SEG=&HFFFF:IF PEEK(14)
      =253 THEN OEF SEG=&H1700:G
      OTO 30
IE 20 DEF SEG=&H1C00
EE 30 FOR I=1 TO 31:READ A$:FOR
      J=1 TO 143 STEP 2
8B 40 POKE K,VAL("&h"+MIO$(A$,J,
      2)):K=K+1:IF K<825 THEN NE
      XT:NEXT
KC 50 8SAVE"chess.b1d",0,825
JG 60 OATA 1EB8311C8E088C16E1008
      926E30088401C8E008C0001E80
      A008E16E1008826E3001FC8FA8
      908008BD9C6875E00C0E2F7C60
      65E0000C606E000008800008FF
      FFFE90B018A854C0002
OJ 70 OATA 855400508A088A8767008
      A904C0088B767008A852C00588
      8876700040650 8B1E2900C6876
      000005B8A8F10002A8D6000C68
      56000C083FF0075523A0E5F007
      C4875118000E643E440
IB B0 OATA E4403A065E007238A25E0
      0803E0F0000741DA04C003A065
      C00752802065400 3A065D00751
      E80F9E57E19FE06E000C3880E5
      F008A0E4C00B80E5C008A0E540
      08B0E5000C33ABD5F00
ND 90 OATA 7EF988BD5F008A9028008
      0C3068A8710002AB55F003A855
      E007C4083FF0174083A855E007
      435C38A804C00028054008A098
      A8767008 03E28000075063C017
      D8B7C083C007C853C07
PF 100 OATA 7481B8852C003C067404
       3CFA750AC6855F002E5A5AE85
       C908A9D4C008A876700C6B767
       00008A0988767003B3E29007
       503E9EFFE47C6854C00148036
       280001FE854C008A904C008A
EK 110 OATA 9F670080 3E280000750D
       B0F8017C1580F8077410E8089
       080F8007D08F6D8D0E3FF971B
       00808D4C00627CCCB3FF00740
       980362B00014FE9A2FEA05C00
       00065D00F8C3803E28000075
HO 120 OATA 5EBA9D4C0080C30A80BF
       6700007523C6B554000AEB38F
       F8A9D4C00B0FB277D1680C314
       808F67000075 08C6B5540014E
       81DFFBA9D4C00B0C309B0BF67
       00007D08C68554000 9EB07FF
DN 130 DATA BA9D4C00B0C30BB0BF67
       00007D08C68554000BEBF1FEC
       38A904C0080C3F6B08F670000
       7523C6855400F6E8DAFE8A9D4
       C0080F8517C16B0C3ECB0BF67
       000075 08C6855400ECEBBFFE
FL 140 DATA BA9D4C00B0C3F7B0BF67
       00007E0BC6B55400F7EBA9FEB
       A9D4C0080C3F5808F6700007E
       08C6855400F5EB93FEC3C6B53
       40000B3008AB70000BBB55400
       EBB0FEFEB53400BA9D3400B0
DB 150 DATA FB087CE8C3C6B5440004
       C6B5340000EB1890C68544000
       8C6B5340004E80B90C6B54400
       08C6B53400008A9D34008AB70
       B00B8B53C00B8855400EB3BFE
       8A854C0002B55400BADBB0BF
NI 160 DATA 670000750ABAB5540002
       B53C00EBDEFE8534008A9D340
       03A9D44007CC8C3C6B5340000
       B3008AB708008BB55400EBFFF
       DFE8534008A903400B0FB0B7C
       E8C30000150CFBEDEBF40B13
KB 170 DATA 0BF7F5090A01F6FF2E09
       0503030100010303 05092EAA0
       16D02BD029A02A702EE02
```

Program 2: IBM Chess (Main Program)

```
KN 10 CO=&H1C00:OEF SEG=&HFFFF:I
      F PEEK(14)=253 THEN CO=&H1
      700:I=1
AB 20 DA=CO+49:DEF SEG=CO:8LOAD"
      chess.b1d",0:IF I THEN POK
      E 3,23:POKE 16,23
JD 30 DEF SEG=DA:GOSUB 690
GF 40 M=40:N=158:K=21
NF 50 POKE 43,1-88:GOTO 180
KO 60 IF C2 THEN 180
NC 70 PDKE 223,0:OEF SEG=CO:SOUN
      D 99,0:CALL ML:DEF SEG=OA
LD 80 IF PEEK(95)<229 ANO PEEK(9
      5)>150 THEN I=0:GOTO 120
8F 90 K1=PEEK(92):K=PEEK(93):SDU
      ND 500,1:GOSUB 1190:GOSUB
      950
QI 100 IF PEEK(95)>99 OR PEEK(95
       )<2B THEN 180
NF 110 I=1
OF 120 X=I+88+PEEK(43):IF I=0 TH
       EN POKE 43,-(PEEK(43)=0)
8N 130 GOSUB 1410:PRINT"Checkmat
       e!  ";
LD 140 IF X/2-INT(X/2) THEN PRIN
       T"White wins.":GOTO 160
IB 150 PRINT"8lack wins."
FD 160 SOUNO 999,9:FOR J=0 TO 20
       0:NEXT
HP 170 SOUND 260,9:FOR J=0 TO 20
       0:NEXT
KG 180 F=0:M=M-8:N=N-3
NB 190 GOSUB 680
NM 200 C$=INKEY$:IF C$="" THEN 2
       00
EP 210 IF LEN(C$)=1 THEN 270
PD 220 C=ASC(RIGHT$(C$,1)):IF C=
       75 ANO M>32 THEN GOSUB 68
       0:M=M-31:K=K-1:GOTD 190
KB 230 IF C=77 ANO M<249 THEN GO
       SUB 680:M=M+31:K=K+1:GOTO
        190
DN 240 IF C=72 ANO N>8 THEN GOSU
       8 6B0:N=N-21:K=K+10:GOTD
       190
DB 250 IF C=B0 AND N<155 THEN GO
       SUB 6B0:N=N+21:K=K-10:GDT
       D 190
BD 260 GOTO 200
QP 270 C=ASC(C$):GDSUB 1400:IF C
       <>13 DR F=0 THEN 360
GF 2B0 POKE 92,K1:POKE 93,K:J=PE
       EK(41):POKE 41,1:POKE 223
       ,1
FF 290 DEF SEG=CD:CALL ML:DEF SE
       G=DA
DD 300 PDKE 41,J:IF PEEK(224)=0
       THEN 320
CK 310 GOSUB 1190:GOSUB 950:GDTO
        60
MG 320 X=PEEK(103+K1):IF (X=6 DR
        X=250) AND ABS(K-K1)=2 T
       HEN GDSUB 1190:GDSUB 950:
       Y=K1:K1=21-70*(X>6)-7*(K>
       K1):K=K+(K>Y)-(Y>K):MM=MM
       -1:GDSUB 1190:PR(MV)=1:GD
       SUB 950:GDTO 60
DD 330 IF PEEK(103+K) THEN 350
NI 340 IF (X=1 DR X=255) AND (AB
       S(K-K1)=9 DR A8S(K-K1)=11
       ) THEN GDSUB 1190:GDSUB 9
       50:K=K+10*(X=1)-10*(X>1):
       MM=MM-1:GDSUB 1190:PR(MV)
       =1:GOSUB 950:GDTO 60
PE 350 SDUND 100,4:F=0:PDKE 43,-
       (PEEK(43)=0):GDTD 200
GA 360 IF F THEN 200
JD 370 IF C<>13 OR PEEK(103+K)=0
        THEN 410
QJ 3B0 IF PEEK(43) ANO PEEK(103+
       K)<7 THEN 400
DH 390 IF PEEK(43) OR PEEK(103+K
       )<7 THEN 410
ND 400 K1=K:F=1:SOUNO 500,1:GOTO
        200
AD 410 S=0
JL 420 IF O(S)=C THEN 450
EN 430 S=S+1:IF S<28 THEN 420
BB 440 GOTO 200
JA 450 IF S>22 THEN SOUNO 500,1:
       LOCATE 1,22:PRINT C$:POKE
        41,VAL(C$):GOTO 200
NA 460 IF S=13 THEN SOUNO 500,1:
       GDSUB 680:M=M+8:N=N+3:GOT
       O 70
IF 470 IF S=14 THEN SOUNO 500,1:
       FOR I=0 TD 70 STEP 10:FOR
        J=0 TD 7:PDKE 124+I+J,0:
       NEXT:NEXT:MX=0:MV=0:MM=0:
       88=0:GOSUB 900:GOTO 40
NN 480 IF S<>15 OR MV=0 THEN 530
LJ 490 SDUNO 500,1:POKE 43,-(PEE
       K(43)=0):GOSUB 6B0:GDSUB
       1200:MM=MM-1:GOSUB 1430
GD 500 IF A8S(PC(MV)-12B)=122 AN
       D A8S(FR(MV)-T(MV))=2 THE
       N GOSUB 1200
FG 510 IF A8S(PC(MV)-128)=127 AN
       O PC(MV+1)=0 AND MV<MX TH
       EN GOSUB 1200
GN 520 GOTO 180
HN 530 IF S<>16 OR MV>=MX THEN 5
       80
FN 540 SOUNO 500,1:POKE 43,-(PEE
       K(43)=0):GOSUB 680:GOSUB
       1210:MM=MM+1:GOSUB 1430
LD 550 IF A8S(PC(MV)-128)=122 AN
       O A8S(FR(MV)-T(MV))=2 THE
       N GDSUB 1210
KC 560 IF A8S(PC(MV)-128)=127 AN
       O PC(MV+1)=0 ANO MV<MX TH
       EN GOSUB 1210
GG 570 GOTO 180
NJ 580 IF S=17 THEN 88=0:GOTO 67
       0
PD 590 IF S=18 THEN 1280
HA 600 IF S=19 THEN 1220
```

```
FE 610  IF S=20 THEN 1340
KL 620  IF S=21 THEN B8=1:GOTO 67
        0
MG 630  IF S=22 THEN SOUND 500,1:
        C2=1-C2
BI 640  IF S>12 THEN 200
HM 650  SOUNO 500,1:IF S>6 THEN S
        =262-S
PI 660  POKE 103+K,S:GOSUB 950:M=
        M-8:N=N-3:GOTO 190
EG 670  SOUND 500,1:MV=0:MM=0:FOR
         I=0 TO 77:POKE I+124,BD(
        I):NEXT:GOSUB 890:GOTO 40
QL 680  PUT (M,N),F,XOR:RETURN
HH 690  KEY OFF:SCREEN 1,0:COLOR
        0,1:CLS
BG 700  POKE 41,1
AB 710  DEFINT P,N,8,R,Q,K,F
LD 720  DIM A(64),C(64),O(27),P(3
        0),N(30),B(30),R(30),Q(30
        ),K(30),F(B2),FR(200),T(2
        00),PC(200),CA(200),PR(20
        0),BO(77)
PP 730  FOR I=0 TO 27:READ D(I):N
        EXT
GJ 740  LINE (0,0)-(29,19),1,BF
MF 750  GET (0,0)-(29,19),A:CLS
HF 760  LINE (0,0)-(29,19),2,BF
AH 770  GET (0,0)-(29,19),C:CLS
QD 780  LOCATE 10,18:PRINT "CHESS
        "
OH 790  LOCATE 12,15:PRINT"John K
        rause"
MM B00  FOR I=103 TO 222:POKE I,7
        :NEXT
GL 810  FOR I=0 TO 77:READ 8O(I):
        POKE I+124,8D(I):NEXT
PM B20  FOR K=0 TO 30:READ P(K):N
        EXT
MP B30  FOR K=0 TO 30:READ N(K):N
        EXT
KB B40  FOR K=0 TO 30:READ B(K):N
        EXT
CD 850  FOR K=0 TO 30:READ R(K):N
        EXT
BM 860  FOR K=0 TO 30:READ O(K):N
        EXT
IP 870  FOR K=0 TO 30:READ K(K):N
        EXT
PL B80  FOR K=0 TO 82:READ F(K):N
        EXT:CLS
JC B90  IF BB THEN POKE 127,6:POK
        E 128,5:POKE 197,250:POKE
         198,251
HM 900  LOCATE 1,5:PRINT"Move#
           Level"PEEK(41)"  To mo
        ve:":GOSUB 1430
EE 910  FOR I=0 TO 7:FOR J=0 TO 7
IM 920  H=70-10*I+J:GOSUB 960:NEX
        T:NEXT
CD 930  FOR I=1 TO B:LOCATE 3*I-1
        +(I>4),2:PRINT 9-I:NEXT
DI 940  GOSUB 1400:RETURN
DO 950  H=K-21:I=INT(H/10):J=H-10
        *I:I=7-I
PA 960  M=31*J+40:N=21*I+11
FF 970  IF INT((I+J)/2)-(I+J)/2 T
        HEN PUT (M-B,N-3),C,PSET:
         GOTO 990
MI 980  PUT (M-B,N-3),A,PSET
HC 990  L=PEEK(124+H):IF I=0 AND
        L=1 THEN L=5:POKE 124+H,L
JF 1000  IF I=7 AND L=255 THEN L=
         251:POKE 124+H,L
PM 1010  IF L>6 THEN L=L-256
OM 1020  ON ABS(L) GOTO 1040,1050
         ,1060,1070,10B0,1090
IL 1030  GOTO 1100
MF 1040  PUT (M,N),P,OR:GOTO 1100
MI 1050  PUT (M,N),N,OR:GOTO 1100
QL 1060  PUT (M,N),8,OR:GOTO 1100
QD 1070  PUT (M,N),R,OR:GOTO 1100
PB 1080  PUT (M,N),Q,OR:GOTO 1100
IG 1090  PUT (M,N),K,OR
PG 1100  IF BB THEN L=-L
KF 1110  IF L>=0 THEN RETURN
PF 1120  ON -L GOTO 1130,1140,115
         0,1160,1170,1180
FD 1130  PUT (M,N),P,XOR:RETURN
DG 1140  PUT (M,N),N,XOR:RETURN
IJ 1150  PUT (M,N),B,XOR:RETURN
IM 1160  PUT (M,N),R,XOR:RETURN
HP 1170  PUT (M,N),Q,XOR:RETURN
BC 11B0  PUT (M,N),K,XOR:RETURN
QM 1190  K2=K:K=K1:MV=MV+1:PR(MV)
         =0:MM=MM+1:MX=MV:FR(MV)=
         K:PC(MV)=PEEK(103+K):POK
         E 103+K,0:GOSUB 950:K=K2
         :T(MV)=K:CA(MV)=PEEK(103
         +K):POKE 103+K,PC(MV):GO
         SUB 1430:RETURN
QM 1200  POKE 103+FR(MV),PC(MV):P
         OKE 103+T(MV),CA(MV):K=T
         (MV):GOSUB 950:K=FR(MV):
         GOSUB 950:MV=MV-1:RETURN
FO 1210  MV=MV+1:POKE 103+T(MV),P
         EEK(103+FR(MV)):POKE 103
         +FR(MV),0:K=FR(MV):GOSUB
          950:K=T(MV):GOSUB 950:R
         ETURN
DM 1220  SOUNO 500,1:GOSUB 1410:I
         NPUT"Save:",N$
KO 1230  ON ERROR GOTO 1420
LA 1240  OPEN N$+".chs" FOR OUTPU
         T AS #1
MC 1250  FOR I=124 TO 201:PRINT #
         1,PEEK(I):NEXT
GM 1260  PRINT #1,PEEK(41),PEEK(4
         3),MV,MX,MM,88,M,N,K,C2
BO 1270  FOR I=1 TO MX:PRINT #1,T
         (I),FR(I),PC(I),CA(I),PR
         (I):NEXT:CLOSE #1:ON ERR
         OR GOTO 0:GOSUB 1400:GOT
         O 200
BC 1280  SOUND 500,1:GOSUB 1410:I
         NPUT"Load:",N$
LA 1290  ON ERROR GOTO 1420
GM 1300  OPEN N$+".chs" FOR INPUT
          AS #1
FJ 1310  FOR I=124 TO 201:INPUT #
         1,J:POKE I,J:NEXT
EM 1320  INPUT #1,X,J,MV,MX,MM,B8
         ,M1,N1,K1,C2:POKE 41,X:P
         OKE 43,J
MI 1330  FOR I=1 TO MX:INPUT #1,T
         (I),FR(I),PC(I),CA(I),PR
         (I):NEXT:CLOSE #1:ON ERR
         OR GOTO 0:GOSUB 900:M=M1
         :N=N1:K=K1:GOTO 190
PC 1340  SOUNO 500,1:X=0:FOR I=1
         TO MX:IF PR(I) THEN 1370
BD 1350  X=X+1:IF X/2-INT(X/2) TH
         EN LPRINT(X+1)/2" ";:GOS
         U8 1380:GOTO 1370
JH 1360  LPRINT"   ";:GOSUB 1380:
         LPRINT
HH 1370  NEXT:LPRINT:GOTO 200
HE 1380  J=INT(FR(I)/10):LPRINT C
         HR$(64+FR(I)-10*J);MID$(
         STR$(J-1),2,1)"-";
MD 1390  J=INT(T(I)/10):LPRINT CH
         R$(64+T(I)-10*J);MIO$(ST
         R$(J-1),2,1);:RETURN
LC 1400  LOCATE 23,6:PRINT"A   8
           C   O   E   F   G   H"
         :RETURN
LE 1410  LOCATE 23,6:PRINT"
                                "
         :LOCATE 23,9:RETURN
LD 1420  GOSUB 1410:PRINT"Error #
         "ERR:RESUME 200
LF 1430  LOCATE 1,10:PRINT INT(MM
         /2+1)" ":LOCATE 1,35:IF
         INT(MM/2)=MM/2 THEN PRIN
         T CHR$(87):RETURN
FE 1440  PRINT CHR$(66):RETURN
MB 1450  DATA 32,80,78,66,82,81,7
         5,16,14,2,18,17,11,109,9
         9,98,102,110,108,115,112
         ,105,116,49,50,51,52,53
CL 1460  DATA 4,2,3,5,6,3,2,4,7
FO 1470  OATA 7,1,1,1,1,1,1,1,1,7
LB 1480  DATA 7,0,0,0,0,0,0,0,0,7
LE 1490  DATA 7,0,0,0,0,0,0,0,0,7
KL 1500  DATA 7,0,0,0,0,0,0,0,0,7
KO 1510  DATA 7,0,0,0,0,0,0,0,0,7
BJ 1520  DATA 7,255,255,255,255,2
         55,255,255,255,7
HO 1530  DATA 7,252,254,253,251,2
         50,253,254,252
BO 1540  DATA 2B,14,0,0,0,0,3B40,
         0
OE 1550  OATA 1612B,192,1612B,192
         ,3840,0,16128,192
AF 1560  DATA 3B40,0,3B40,0,1612B
         ,192,-256,240
OG 1570  OATA -256,240,0,0,0,0,12
         B
PK 15B0  DATA 28,14,3,0,-16381,0,
         -1021,0
LC 1590  DATA-241,192,-244,240,-2
         41,240,-241,252
PL 1600  DATA -193,252,-124B1,255
         ,3B52,255,16128,255
KF 1610  DATA -256,255,-253,255,-
         253,255,-253
PB 1620  DATA 2B,14,-4096,240,-40
         96,240,-1021,252
CF 1630  OATA -253,60,-253,204,-2
         53,204,-253,204
KK 1640  OATA -256,240,-16384,48,
         -256,240,-16384,48
LL 1650  OATA -193,-16129,-3841,-
         3841,192,12288,-253
OP 1660  DATA 28,14,16143,207,161
         43,207,-241,255
IJ 1670  DATA 3,12,-253,252,-253,
         252,-253,252
MO 1680  DATA -253,252,-253,252,-
         253,252,3,12
EG 1690  DATA -241,255,-193,-1612
         9,-193,-16129,-193
CE 1700  DATA 28,14,-16384,192,-1
         6384,192,-16384,192
FA 1710  DATA -16192,-16192,-3133
         ,-16144,-3277,243,-3277,
         243
ED 1720  DATA -193,255,12,12,-241
         ,252,-3313,252
KI 1730  DATA -241,252,12,12,-241
         ,252,0
CD 1740  OATA 28,14,-256,192,-133
         12,192,-3268,207
KP 1750  DATA -13057,-16129,-1,-1
         6129,-16129,-16129,-3265
         ,255
FP 1760  DATA -193,255,12,12,-241
         ,252,-3313,252
KE 1770  DATA -241,252,12,12,-241
         ,252,0
CJ 1780  DATA 60,20,-1,-1,-1,-384
         1,-1,-1
GB 1790  DATA -1,-3841,252,0,0,-4
         093,252,0
DF 1800  DATA 0,-4093,252,0,0,-40
         93,252,0
EI 1810  DATA 0,-4093,252,0,0,-40
         93,252,0
EL 1820  DATA 0,-4093,252,0,0,-40
         93,252,0
ED 1830  DATA 0,-4093,252,0,0,-40
         93,252,0
EB 1840  DATA 0,-4093,252,0,0,-40
         93,252,0
EE 1850  DATA 0,-4093,252,0,0,-40
         93,252,0
EH 1860  DATA 0,-4093,252,0,0,-40
         93,252,0
II 1870  DATA 0,-4093,-1,-1,-1,-3
         841,-1,-1
IF 1880  DATA -1,-3841,0
```

©

Commodore Bootstrapping

Jim Butterfield, Associate Editor

Large programs are often divided into several parts and started up by a separate program called a bootstrap. This article explains how the technique works and provides a simple demonstration. The demo programs run on the Commodore 64, VIC-20, 16, Plus/4, 128 (in 64 mode), and PET/CBM, and require a disk drive.

Many complex programs—especially commercial software packages—appear on disk or tape as a collection of files. The program is broken into several pieces, and each file is one of the pieces. It's the job of a *bootstrap* program (often called a *boot)* to put all these pieces together. This makes your job easier: Just load the boot program and enter RUN. The boot brings in the other programs and gets everything going for you.

When you see a cluster of programs with similar names on a disk, look for one with BOOT in the name. That's the one to load and run. For instance, you might see these filenames in a disk directory:

```
GAME.BOOT
+GAME.SCREEN
+GAME.MUSIC
+GAME.SPRITES
+GAME.ML
+GAME.MAIN
```

In this case, you run GAME.BOOT. The boot loads each of the remaining files in turn: +GAME.SCREEN, which contains a drawing of a high-resolution screen; +GAME.MUSIC, a tune that plays during the game; +GAME.SPRITES, which contains pictures of moving objects; +GAME.ML, a machine language routine used by the main program; and finally, +GAME.MAIN, which is the actual game program. When the bootstrap program has finished its job, often it erases itself from memory.

Notice in the above example how all the filenames other than the bootstrap start with a nonalphabetic character. The computer doesn't care what the filenames look like; the symbols are a signal to you, the human part of the system, that you shouldn't load these programs directly.

In other cases, you don't get any hints from the filenames. The word BOOT doesn't appear in any filename, and the names are not distinguished by any special symbols. With a commercial program, you could try LOAD "*",8,1 to see if this starts a bootstrap sequence. If all else fails, you may have to try desperate measures: Read the instructions.

A Little History

Early computers had no Read Only Memory. The marvelous ROM that computers now use to store "canned" instructions didn't exist. When the computer was turned on, it knew nothing—not even how to load a program. Thus, early computer users were faced with a chicken-and-egg paradox: In order to load a program, they needed a program in the computer that told it how to load. How did they get this first program in? Sometimes toggle switches were used to enter individual bytes. Sometimes the computer could read a punched card and transfer a tiny program from the card into its memory.

Whatever the method, one thing was certain: The first program would be very small, containing just enough instructions to do the simplest possible loading job. And the first program to be loaded would usually be a bigger and better loading program. You had to start with a tiny loading program whose job was to bring in a bigger loading program. It seemed as though the computer was coming into action by pulling itself up "by its own bootstraps." And the term *bootstrap* came to signify any program whose job is to bring in a larger program.

Once you open the door to program-loading programs, new possibilities arise. For example, a bootstrap program can bring in several disconnected modules, each of a different type (a screen, a main BASIC program, a machine language routine, and so on). Since the modules may load into different memory areas, it's usually far easier to create them as separate files rather than paste them into one big package that loads as a single file.

A bootstrap program can also reconfigure the computer. To make room for a high-resolution graphics screen or extra sprite definitions, you may need to change the locations where BASIC starts and ends. The boot program can reconfigure BASIC memory, then load the main BASIC program into the newly defined area.

The bootstrap can make changes to allow for a particular model of computer. If the boot program finds it is running in an 80-column machine, it might decide to load an 80-column program module instead of the 40-column one. Or, the boot could let the user decide what modules to load, depending on what peripherals are in use. Thus, the program might ask if the user has a color or black-and-white monitor, or call for the identity of any printer that is connected.

Writing A Simple Boot

Let's write a small program that uses a bootstrap technique. We'll make the program do a simple task: read a sequential file from disk. If you don't happen to have a sequential file on disk, you can create a short one called XFILE by typing the following statements in direct mode (without a line number).

```
OPEN 8,8,8,"Ø:XFILE,S,W"
PRINT#8,"HELLO THERE"
PRINT#8,"GOODBYE NOW"
CLOSE 8
```

Now for the program itself. Here's the plan: We'll put a main program in BASIC's usual memory area. In another area (the cassette buffer), we'll put a machine language (ML) routine that reads the file quickly and displays it on the screen. Finally, we'll need a bootstrap program to install the other two modules. We'll be using several advanced techniques, including machine language programming, program overlays, and dynamic keyboard. If you haven't seen them before, don't worry. There's no space here to explain the techniques in detail, but you can still run the programs and enjoy the view.

First you need to put an ML routine on disk. The following program is not an ML routine itself, but a generator program that creates one for you. Type in and save the program, then run it. *(Be sure to type the semicolon at the end of line 220.)* This program puts a short machine language program named "+ML" on your disk. If the computer prints ** ERROR **, you've made a typing mistake in the DATA statements. After you correct the error in the generator program and resave it, scratch the incorrect ML file by typing OPEN 15,8,15,"S0:+ML": CLOSE 15. Then reload the generator program and run it again.

If you have a Commodore 128, you can type in and save the programs in 128 mode, but before running the boot you must switch to 64 mode as explained below. The value of 144 in line 150 is correct for the VIC-20, Commodore 64 (and 128 in 64 mode), 16, and Plus/4. It needs fixing for the PET/CBM, but we'll let the boot program do that.

```
100 DATA 6Ø,3
11Ø DATA 162,1
12Ø DATA 32,198,255
13Ø DATA 32,228,255
14Ø DATA 32,21Ø,255
15Ø DATA 166,144
16Ø DATA 24Ø,246
17Ø DATA 76,2Ø4,255
18Ø OPEN 4,8,4,"Ø:+ML,P,W"
19Ø FOR J=1 TO 2Ø
2ØØ READ X
21Ø T=T+X
22Ø PRINT#4,CHR$(X);
23Ø NEXT J
24Ø CLOSE 4
25Ø IF T<>3Ø54 THEN PRINT "**
    {SPACE}ERROR **"
```

Creating The Main Program

The BASIC program is quite straightforward. Type NEW and enter:

```
1ØØ PRINT "NAME OF SEQUENTIAL
    {SPACE}FILE":INPUTN$
11Ø OPEN 1,8,2,N$
12Ø SYS 828
13Ø CLOSE1
```

Now save this program by typing SAVE "0:+BASIC",8 so that the boot program can call it up when needed. *Do not try to run this program yet.* First we have to put the machine language routine it uses into memory.

Creating The Bootstrap

Type NEW again. Since the boot program varies slightly depending on the computer, we'll take care of the differences in the first line of the program. Enter line 100 as listed below for your computer.

For the 64 and VIC-20 (or 128 in 64 mode):

```
1ØØ DATA 144,198,631
```

For the Commodore 16 or Plus/4:

```
1ØØ DATA 144,239,1319
```

For the PET/CBM:

```
1ØØ DATA 15Ø,158,623
```

The three values in line 100 represent the memory locations of the computer's status variable (ST), keyboard buffer counter, and keyboard buffer, respectively. The first value adjusts the ML program to work on different machines. The other two are used to load the main BASIC program with the dynamic keyboard technique. After you enter line 100, type in the following lines as well:

```
11Ø IF X=1 GOTO 2ØØ
12Ø X=1
13Ø LOAD"+ML",8,1
14Ø STOP
```

We're using a program overlay technique here. The computer never reaches line 140, since the boot program restarts at its first statement with all variable values intact after the LOAD in line 130. Since the variable X equals 1 on the second pass, the computer leaps ahead to the rest of the program at line 200. The technique is called program overlay because it was designed to allow a second BASIC program to be loaded over an existing program while maintaining variable values. Whenever a LOAD command is executed within a program, whatever BASIC program is in memory after the LOAD is finished will begin running at its first line. We're not actually using an overlay here, since the machine language program doesn't overwrite the BASIC boot program in memory, hence the need for using X to skip the LOAD on the second pass. Without it, the program would do nothing but LOAD again and again.

Now enter the following lines, which adjust the ML program to run on different machines.

```
2ØØ READ A,B,C
21Ø POKE 84Ø,A
```

Loading the ML required a special overlay technique. Loading the BASIC program is even trickier. Since BASIC programs normally load into the same space, the new program will destroy the bootstrap as it comes in. There are several ways we can cope with this. Perhaps the easiest is to use the dynamic keyboard technique. Here goes:

```
22Ø D$=CHR$(17)
```

```
230 R$=CHR$(147)+D$+D$+D$+"LOA
    D"
240 N$=CHR$(34)+"+BASIC"+CHR$(
    34)
250 PRINT R$+N$+",8"+D$+D$
260 PRINT D$+D$+"RUN"+CHR$(19)
270 POKE B,2:POKE C,13:POKE C+
    1,13
```

If you've never used the dynamic keyboard technique, the above lines may look confusing. Briefly, we are telling the computer to type two commands on the screen for us. You'll see the commands when the program runs:

```
LOAD "+BASIC",8
RUN
```

The commands are carefully arranged on lines 3 and 8 of the screen. If you pressed RETURN twice—assuming the cursor was in the right place—the commands would execute, loading and running the program named +BASIC. But the boot program can press RETURN for us by putting RETURN characters, CHR$(13), into the keyboard buffer. This is a familiar trick for making Commodore computers do things that would otherwise be difficult.

Our bootstrap program is complete. Save it on disk with the name BOOT. *Be sure to save a copy of the program before you run it,* since it erases itself from memory after performing its work. (Users of the 128 *must* switch to 64 mode before running the program. Type GO64 and then enter Y at the prompt.) You should now have the following files on your disk:

BOOT (the boot program you just entered)
+ML
+BASIC
The sequential file you wish to read (**XFILE**, for example)

When you run the boot program, it loads in the ML and BASIC modules and starts things up. You'll be asked for a filename (enter XFILE if you created the sample file as shown above). After the program is finished, you can look at another file without using the boot again. Since everything's in place, just enter RUN.

This simple demonstration only hints at what a bootstrap program can do. The small but mighty bootstrap can call together many program elements to create an elegant and effective software package. ©

Atari Animation With P/M Graphics

Part 1

Robert J. Powell

Here's an easy-to-grasp explanation of how to use the Atari computer's built-in system for advanced graphics animation. This month, Part 1 takes you step by step through the fundamentals of setting up player/missile graphics in BASIC. It's intended for those with an intermediate knowledge of BASIC programming.

One of the reasons you probably bought an Atari computer was for its fine graphics capabilities. By now, maybe you've tried to write some programs with graphics and discovered that it takes considerable work to achieve the special effects you've admired in commercial software. Smooth animation seems impossible with ordinary character graphics, and moving any object across the screen using BASIC is difficult and often disappointingly slow.

The alternative is that mysterious Atari feature known as *player/missile graphics.* With P/M graphics, you can create shapes in any color and move them smoothly around the screen with relative ease. You can simulate three-dimensional movement by making some shapes pass over or beneath other shapes and the screen background. You can even detect when a shape has collided with another shape or with anything else on the screen. P/M graphics is the key to sophisticated animation on Atari computers.

Unfortunately, too many people are intimidated by P/M graphics. Although it isn't the Atari's easiest to use feature, it isn't the most difficult, either. The mystery surrounding P/M graphics started soon after the original Atari 400 and 800 computers were introduced in 1979. It was obvious from early commercial games like *Star Raiders* that some innovative graphics were involved, but Atari didn't even mention the feature in any of its manuals. Indeed, the first explanation of how P/M graphics works didn't appear until January 1981, when Atari programmer Chris Crawford wrote an article entitled "Player/Missile Graphics with the Atari Personal Computer System," which appeared in COMPUTE!. Until then, most programmers were in the dark.

A number of magazine articles and books followed, most notably *De Re Atari* by Crawford and his colleagues at Atari. But since the latest generation of Atari XL and XE owners has missed all this history, it's time for another look at P/M graphics and how it can help you add the professional touch to your programs.

A Layer Of Cellophane

First of all, P/M graphics isn't part of BASIC; there aren't even any Atari BASIC commands or keywords for handling P/M graphics. Instead, P/M graphics is built into the hardware of the computer, specifically the dedicated graphics chips unique to the Atari. Therefore, all P/M manipulation in BASIC must be done with PEEK and POKE statements.

A good way to think of P/M graphics is as a second video image overlapped onto the regular screen, like a layer of colored cellophane. That's why P/M objects can seem to travel over or behind other screen objects without erasing or disturbing them.

This system is known as *sprite graphics* on most other computers, such as the Commodore 64 and TI-99/4A. On these machines, each movable object is called a sprite; the Commodore can display up to eight at a time without special tricks, and the TI can display up to 32. Atari P/M graphics, an earlier system, consists of eight movable objects, but they're a little different than sprites. On the 64 and TI, sprites are all the same size and are roughly square (although they can be redefined as any shape, of course). On the Atari, there are four full-sized objects called *players* and four miniature objects called *missiles.* If you want, the four missiles can be grouped together to form a fifth player. And instead of being square, players and missiles are narrow strips taller than the height of the screen.

If you've never seen these strips, don't be surprised. Most programs that use P/M graphics render all but a small part of the strip invisible on the screen. The small visible part is the player or missile object you actually see. Its shape is determined by numbers POKEd by the program into a section of memory called *P/M graphics memory.* It's up to your program to set aside and protect this memory when it runs. When your program fills this memory with zeros, the whole P/M strip becomes invisible. By POKEing a few nonzero numbers into P/M memory, your program defines the shape of the visible part of the strip. This shape could be an alien, a spaceship, a cursor for a spreadsheet, or almost anything you want.

In P/M memory, each player strip is eight bits (one byte) wide, and each missile strip is two bits wide. (That's why grouping together the four two-bit missiles results in a fifth player.) All the strips are either 128 or 256 bytes tall (as described below) and extend off the visible screen in both directions.

Later, we'll explain how to determine which numbers to POKE to redefine the strips into your own shapes.

P/M Memory

Once defined, players and missiles can appear in any graphics or text mode and can be quickly moved about the screen without affecting the background graphics or text. Each player can be a different color, and P/M colors can be different than the regular screen colors—thus allowing more simultaneous colors than are normally available. With a few PEEKs, you can check for collisions between players, players and missiles, and players and screen objects (including characters). Before creating a player, let's take a look at how P/M memory is organized.

Your program must set up P/M memory to store the shape data for players. The amount of memory you set aside depends on the degree of P/M resolution desired. Two resolutions are available: single scan-line and double scan-line (a *scan-line* is the thinnest horizontal line visible on your video screen). Single-line resolution allows more detailed shapes but requires twice as much P/M memory. A single-line player is 256 bytes tall and a double-line player is 128 bytes tall. Single-line resolution requires a total of 2K, or 2,048 bytes; double-line resolution requires a total of 1K, or 1,024 bytes.

To protect P/M memory against intrusions, it's generally established near the top of user RAM just below screen memory. Another requirement is that P/M memory must start on an address that is a multiple of eight pages (2K) for single-line resolution or a multiple of four pages (1K) for double-line resolution. (A *memory page* equals 256 bytes.)

The accompanying figure shows a map of P/M memory. By custom, the starting address of P/M memory is assigned to the variable PMBASE. Since the exact memory address of PMBASE varies according to how much RAM is in the computer, which graphics mode you're using, and other factors, the map shows all other addresses as relative offsets from PMBASE. For single-line resolution, the missile data area occupies 256 bytes starting at PMBASE+768. Player data starts at PMBASE+1024 and requires 256 bytes for each player (numbered 0 through 3). For double-line resolution, all these offsets would be halved, since only half as much memory is required. Missile data would start at PMBASE+384 and player data would start at PMBASE+512.

A Bunch Of POKEs

For an example, let's write a program to set up single-line resolution P/M graphics. This requires a bunch of POKEs which may look confusing. Even if you don't fully understand the purpose of the POKEs, however, you can still use them in your programs.

First, you have to determine the number of memory pages to the starting address of P/M memory, or PMBASE. To do this, you use a memory address called RAMTOP. Logically enough, RAMTOP stores the address of the top of available RAM. That is, the computer looks at RAMTOP to calculate how much free memory is available and won't let BASIC use any memory above RAMTOP. By POKEing a lower value into RAMTOP, you can make the computer think there is less RAM and therefore free up some memory above RAMTOP (just as lowering your ceiling would create more room in your attic). The extra RAM freed up by this method is ideal for P/M memory because it's relatively safe from interference.

The value stored in RAMTOP is the number of memory pages available. How far should you lower RAMTOP? Remember that 1K is required for double-line resolution P/M graphics and 2K is required for single-line resolution P/M graphics. Since we're using single-line resolution in our example, we need to protect 2K (2,048 bytes) for P/M memory. That means we must subtract eight pages from the value in RAMTOP (8*256=2,048). The address for RAMTOP is 106 decimal, so the statement looks like this:

```
10 POKE 106,PEEK(106)-8
```

Second, you must store this new page number for RAMTOP in the *P/M base register* at memory location 54279:

```
20 POKE 54279,PEEK(106)
```

Third, select your graphics mode with the usual GRAPHICS statement, then establish the actual starting address for PMBASE. Let's stick with ordinary text mode and make the screen background black for maximum contrast:

```
30 GRAPHICS 0:SETCOLOR 2,0,0
40 PMBASE=PEEK(106)*256
```

Finally, two more POKEs are required to enable the *Direct Memory Access control register* (559 decimal) and another address which turns on P/M graphics (53277 decimal):

```
50 POKE 559,62
60 POKE 53277,3
```

(Note that for double-line P/M resolution, line 50 would be POKE 559,46.)

P/M graphics memory is now set up and activated. Before you can run the program and actually see the players, though, you have to define some shape data, assign colors, and position them on the visible part of the screen. These tasks require a few additional POKEs.

Revealing The Strips

Let's assign the colors first. There aren't any BASIC statements like COLOR or SETCOLOR for P/M graphics, so you have to POKE color values into certain memory locations instead. Each of the four players has its own color location, or *player color register*. These memory locations are 704 for player 0, 705 for player 1, 706 for player 2, and 707 for player 3. (Incidentally, the missiles lack independent color control, so missile 0 takes the same color as player 0, missile 1 takes the same color as player 1, etc.)

To determine which number to POKE into the player color registers, consult the accompanying table of Atari color numbers and use this formula:

Atari Color Numbers

0 Gray	8 Blue
1 Gold	9 Light blue
2 Orange	10 Turquoise
3 Red-orange	11 Green-blue
4 Pink	12 Green
5 Purple	13 Yellow-green
6 Red-orange	14 Orange-green
7 Blue	15 Light orange

P/M color = color number * 16 + luminance

Luminance means brightness; this should be an even number from 0 to 14. To make player 0 appear medium pink, you could POKE 704,72 (72=4*16+8). To make player 3 appear dark green, POKE 707,13*16+4. (The exact hue may vary according to how your TV or monitor is adjusted.) For our example program, we'll make the players red, green, light blue, and dark blue:

```
70 POKE 704,68:POKE 705,198:POKE
   706,168:POKE 707,148
```

Next, we want to make sure the player strips are positioned where we can see them. In addition to a color register, each player also is controlled by a *horizontal position register.* This is a memory address that determines each player's horizontal location. The registers are 53248 for player 0, 53249 for player 1, 53250 for player 2, and 53251 for player 3. You can POKE any value into these registers from 0 to 255; lower values position the player to the left, and higher values position the player to the right. However, values less than 45 begin moving the player off the left edge of the visible screen, and values greater than 205 begin moving the player off the right edge of the screen.

For this example, let's group all four players together near the right edge of the screen:

```
80 POKE 53248,160:POKE 53249,170:
   POKE 53250,180:POKE 53251,190
```

Finally, to make the player strips visible, we must fill P/M memory with shape data. For now, let's not worry about creating a fancy shape such as a spaceship. Instead, we'll reveal the players as they really are by completely filling P/M memory with 255:

```
90 FOR X=PMBASE+1024 TO
   PMBASE+2048:POKE X,255:
   NEXT X
```

Now run the program. In a few seconds, you'll see the four player strips appear on screen as line 90 fills P/M memory with the shape data.

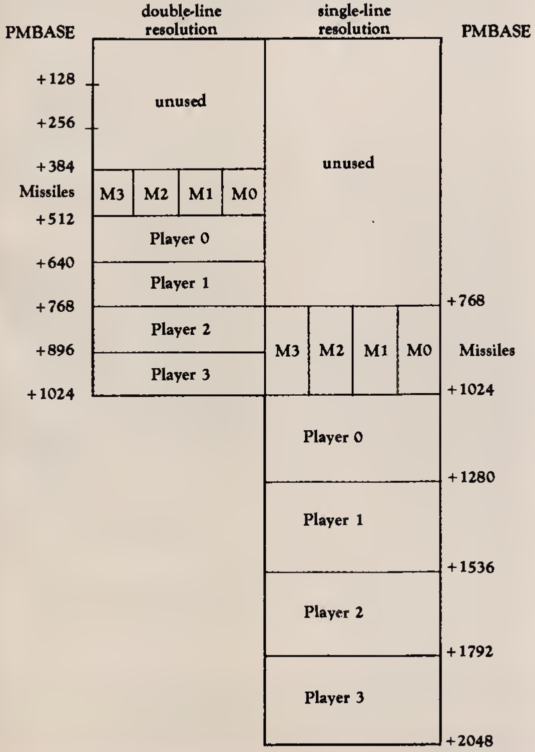
A Few Experiments

After the program stops, the READY prompt reappears and the four players remain on the screen. This is an ideal time to observe how P/M graphics works. Try these experiments:

- Type LIST. Notice how the program listing on the screen overlaps the players.
- Press SHIFT-CLEAR or CTRL-CLEAR. This clears the program listing off the screen but leaves the players undisturbed. P/M graphics, remember, are independent of regular screen graphics and text.
- In direct mode (without a line number), change the color of player 0 by POKEing a different value into the player 0 color register—for example, POKE 704,250. Also change the colors of players 1, 2, and 3 by POKEing color registers 705, 706, and 707.
- In direct mode, relocate player 0 to the left side of the screen by POKEing a lower value into the player 0 horizontal position register—say, POKE 53248,60. Relocate the other players, too, by POKEing their horizontal registers. Make a player disappear from the visible screen by POKEing a value from 0 to 45 or 205 to 255. Try stacking two players atop each other by POKEing the same value into their horizontal registers, and observe which one has display priority.

Next month, we'll show additional ways to manipulate P/M graphics and also how to transform the player strip into a shape of your own design. ©

P/M Graphics Memory Map



All About IBM Batch Files
Part 1

G. Russ Davies

IBM batch programs provide a convenient way to carry out a series of DOS (Disk Operating System) commands at once. This month we'll cover some batch programming fundamentals. Part 2 will show how to add multiple-option menus, color, and graphic displays to batch programs.

In IBM parlance a *batch* program is simply a disk file containing a series (batch) of DOS commands. The batch file executes these commands in sequence, just as if you manually typed them yourself. Batch files are identified with the .BAT filename extension. The most familiar example of a batch program is AUTOEXEC.BAT, used to issue startup commands to configure the system to your liking. Here's what a typical AUTOEXEC.BAT file might contain:

```
MODE CO80
DATE
TIME
CHKDSK
BASICA MENU
```

The first four commands in this batch file are familiar DOS commands to set the display mode to 80 columns, let you input the date and time, and analyze the disk directory. (Note that if the AUTOEXEC.BAT file doesn't include DATE and TIME, the system doesn't ask for date and time inputs when it boots.) The last command activates BASICA, then loads and runs a BASIC program named MENU. A file named AUTOEXEC.BAT differs from other batch files only in that it runs automatically when you turn on the system.

To run a batch program that doesn't automatically run, simply enter the filename at the DOS prompt (you can leave off the .BAT extension). This tells DOS to load the batch file from disk and carry out each of its commands in order. For instance, to run a program named SETUP.BAT you would type SETUP after the DOS prompt and press Enter.

This article presents several example batch programs. Since these are *not* BASIC programs, don't try to enter them with the "IBM Automatic Proofreader." The DOS manual explains how to type in short batch programs using the COPY CON: command from DOS. However, for any batch program longer than a few lines, it's easier to use a word processor or any text editor that creates standard ASCII files. Most commercial programs are suitable. You can also use the EDLIN program (on the *DOS Supplemental Programs* disk), though it lacks the convenient editing features of word processors.

Chains And Parameters

In the AUTOEXEC.BAT example above, the batch program ends by loading BASIC and running a BASIC program. A batch program can also end by returning control to DOS, or by running a second batch program (permitting you to "chain" two or more programs together). For instance, ending a batch program with SECOND causes the system to load and run the batch program named SECOND.BAT. You can also use COMMAND /C to run one batch program from within another: For example, COMMAND /C SECOND runs SECOND.BAT.

Passing parameters (information) to a batch program is straightforward. Simply include the needed information after the filename when running the program. For example, typing FIRST JULIA 123 runs the FIRST.BAT program and passes two parameters to it: a string (JULIA) and a number (123). In much the same way, one batch program can pass parameters to another. Let's use an example to demonstrate parameter passing in chained programs. Enter the following batch program and save it to disk with the filename FIRST.BAT:

```
ECHO OFF
ECHO FIRST.BAT USES FIRST P
ARAMETER: %1
ECHO PASSES %2 AND %3 TO SE
COND.BAT
REM SECOND %2 %3
```

Now enter the following program and save it with the filename SECOND.BAT:

```
ECHO SECOND.BAT USES SECOND
 PARAMETER: %1
```

```
ECHO PASSES %2 TO THIRD.BAT
THIRD %2
```

Finally, enter the following program and save it with the filename THIRD.BAT:

```
ECHO THIRD.BAT USES THIRD P
ARAMETER: %1
```

At this point you have three batch programs, all of which expect parameters. To run the programs, enter FIRST followed by any three strings or numbers. Be sure to separate each parameter with a space. For instance, you might enter FIRST PARAM/ONE &H464 IBMBIO.COM. The FIRST.BAT program takes in all three parameters, processing the first (displaying it in an ECHO statement) and passing the other two when it runs SECOND. SECOND.BAT processes the second parameter and passes the third to THIRD.BAT.

As shown in these examples, batch programs use dummy parameters (% followed by a digit from 0–9) to mark the spot where the real parameter is expected. When you run a batch program, each dummy parameter is replaced by actual data *in the order it is received.* Thus, the FIRST.BAT program above uses %1 to signify the first parameter, %2 to represent second, and so on. Dummy parameter %0 can only be replaced by a drive designator (A or B) and filename: Don't use it unless you want to pass such information.

Be sure to keep the dummy parameter numbers straight when chaining batch programs. The dummy number represents the order *in which that program receives the data.* In the example above, FIRST.BAT received three parameters, which it represents with the three dummies %1, %2, and %3. SECOND.BAT receives two parameters, using %1 to signify the first parameter it receives, and %2 to represent the second. Likewise, THIRD.BAT uses %1 to represent its single parameter. (Note that THIRD.BAT can't use %3 for the dummy. Though you, the programmer, may think of this parameter as the "third," it's the first one that THIRD.BAT receives.)

Batch Commands

In addition to ordinary DOS commands, a batch program may include the following special batch commands: ECHO, FOR, GOTO, IF, SHIFT, PAUSE, and REM. ECHO ON causes DOS commands to be displayed as they're performed in a batch program; ECHO OFF turns off the display. As you saw above, ECHO can also display messages. GOTO is discussed in Part 2 of this article. REM lets you include remarks, and SHIFT is used when more than ten parameters are passed at one time.

The remaining commands (FOR, IF, and PAUSE) permit loops, conditional tests and limited user input. The short file copying program listed below demonstrates all three of these commands. Enter the program as listed, saving it with the filename COPYUNQ.BAT (or any other name ending in .BAT).

```
ECHO off
REM---------------------------
 -----------------------------
 -----------
REM name: COPYUNQ.BAT
REM syntax: COPYUNQ
 source-drive-letter
 target-drive-letter  (no
 colons)
REM purpose: Only unique files
 are copied from source to
 target disk
REM---------------------------
 -----------------------------
 --------------
%1:
FOR %%f in (*.*) DO IF exist
 %2:%%f ECHO %%f WILL NOT BE
 COPIED
PAUSE READY TO BEGIN COPIES,
FOR %%f in (*.*) DO IF not
 exist %2:%%f COPY %1:%%f %2:
 /V
%2:
```

The COPYUNQ.BAT program automatically copies files from a source disk to a target disk, copying only those files that don't already exist on the target disk. This ensures that existing files are not replaced, an improvement over DOS's COPY command, which would write over any like-named files on the target disk. To run this program, enter its name followed by the letter of the source drive and the letter of the target drive. Colons are not required after the drive letters. For instance, you would enter COPYUNQ.BAT A B when drive A holds the source disk and drive B holds the target disk. The program displays the names of files that are not copied.

FOR And IF

COPYUNQ.BAT offers a good demonstration of FOR and IF, which work very differently than their BASIC equivalents. Since a FOR statement can't contain another FOR statement, you can't use nested FOR loops (one FOR loop enclosed by another). FOR statements take the following general form:

FOR *%%variable* IN *(set)* DO *DOS command*

The *set* value after IN represents a group of files and must be some variation of a filename and extension. This parameter determines which disk files the FOR loop will affect. Since the pattern-matching symbols * and ? can be used, you may define this group to be very broad or very selective. The program shown above uses the statement IN (*.*) to affect the broadest possible group: every file on the disk. In other cases, you might use IN (*.BAS) to affect all files ending with .BAS, IN (ABC*.*) to affect all files starting with ABC, and so on.

The first FOR statement in COPYUNQ.BAT (FOR %%f IN (*.*) DO) affects every file on the disk. As the FOR loop executes, the variable %%f represents each filename in order. Translated into plain English, this statement means "cycle through every filename on the source disk, using %%f to represent each filename in turn."

IF can perform only a few tests. One of these (IF EXIST *filename*) tests whether a given file exists on the disk. Now you can understand the second part of the FOR statement (IF EXIST %2:%%f). The %2 parameter is a dummy, replaced by the second drive letter you entered when running the program. And the variable %%f is replaced by actual filenames when the program runs. In plain English, this statement means "if the current filename exists on the disk in the target drive...."

Batch programs don't have the equivalent of BASIC's THEN statement (THEN is implied). But in other respects IF processing works much as it does in BASIC. Statements that come after the IF test (on the same line) are performed when the IF test is true, and skipped when the test is false. Consequently, in COPYUNQ.BAT, the ECHO command (which prints "*filename* WILL NOT BE COPIED") executes only when the file in question exists on both the source and target disks.

Once you understand that much of COPYUNQ.BAT, the rest is not hard to decipher. PAUSE makes the system stop and display the message "Strike any key when ready." This is the only batch command that allows user input. Unfortunately, your choices are severely limited: You can continue only by pressing a key (perhaps after changing disks, etc.) or end the program by pressing Ctrl-Break. In Part 2 of this article, we'll show how to expand this number of options.

NOT And ERRORLEVEL

The second FOR line in COPYUNQ .BAT has a FOR loop and an IF test very similar to the first. However, in this case NOT reverses the logic of the IF test. When the named file *does not exist* on the target disk, the IF test is true and the file is copied.

In addition to testing EXIST (with or without NOT), IF can test two conditions: the equality symbol (==) and ERRORLEVEL. The equality symbol tests whether two strings are identical. ERRORLEVEL is always a number, ordinarily used to pass information from one program to another (indicating whether the first worked successfully and thus set ERRORLEVEL to the expected value). ERRORLEVEL is discussed further in Part 2.

As shown in these brief examples, batch programs can be very powerful: IF lets you pick only the files you want, and FOR lets you repeat commands until the whole task is done. In one sense, the lack of opportunity for user input is an advantage: The entire procedure is automated, and you don't need to understand anything except how to type in the program name. On the other hand, batch programming can seem rigid, limiting, and visually quite dull. Part 2 improves on that situation, offering program examples and a routine that adds colorful graphic displays and multiple-option menu selection to batch programs. ©

News & Products

Commodore Memory Expansion, Interface

Cardco, Inc., has announced *S'more* (Super Memory Optimized RAM/ROM Expansion), a cartridge utility for the 64 which allows more than 60K RAM for programming and adds over 60 new and enhanced BASIC commands and functions. The memory increase is not restricted, and can be used for arrays, variables, and BASIC programs which would normally overload a Commodore 64. *S'more* provides such programming aids as CATALOG (view disk directory), AUTO (line numbering), FIND, CHANGE, TRACE, DUMP, KEY (define function keys), and others.

Function keys are preprogrammed, but can be redefined. For example, F2 runs the current program in memory, F3 reads and displays the disk drive error channel, and F7 displays the current disk directory. The suggested retail price is $69.95. Cardco also plans to introduce the *S'more BASIC Compiler* for $39.95.

Also recently introduced is G Whiz, an improved version of Cardco's +G printer interface, which allows Commodore computers to be hooked up to virtually any Centronics printer. Additional features include faster printing speed (up to 18 times faster with many dot matrix printers), and increased speed on high-resolution screen dumps. The interface also comes with two character sets and open access to DIP switches. The interface attaches directly to the parallel port, eliminating the ribbon connector. Suggested retail price is $69.95.

Cardco, Inc., 300 S. Topeka, Wichita, KS 67202
Circle Reader Service Number 232.

IBM, ST Expert Investment Help

Batteries Included has introduced the first product in its Integral Solutions line of productivity software. The *Isgur Portfolio System* was designed by Lee Isgur, a well-known Wall Street analyst and first vice president of Paine-Webber, Inc. The program allows both casual and professional investors to track up to ten portfolios, each with 50 stocks and 15 separate holdings. With a ten-megabyte hard disk, storage capacity jumps to 1,000 portfolios, with more than 2,000 stocks and 600 holdings of each.

Special tracking and advisory features help determine how and when to raise money, when to sell holdings, and how to prepare for changes in the status of holdings. Built-in telecommunications functions put the user online with major telecommunications services at the touch of a key or two.

The *Isgur Portfolio System* is available for the Atari 520 ST and IBM PC for $249.95.

Batteries Included, 30 Mural St., Richmond Hill, Ontario, Canada L4B 1B5
Circle Reader Service Number 233.

Home Control Package

The X-10 Powerhouse interface is a freestanding controller for lights, heating, cooling, security devices, and other appliances, which you preset with your computer by following simple software-driven onscreen icons representing controllers for each room of your home or business. Available initially for the Apple II series, the system is scheduled to be available for the Commodore 64/128 in September and the IBM PC/PCjr in October.

The Powerhouse lets you control up to 72 lights and appliances plugged into System X-10 modules, which in turn are plugged into your home's electrical outlets. To program the Powerhouse interface, you use a joystick to graphically "install" lights and appliances in each room in positions which correspond to the actual locations in your own home. Once programmed with your computer, the system operates independently. X-10 modules can be purchased at electronics stores. The Powerhouse interface sells for approximately $125, while the appropriate software and connecting cable retails for an additional $25.

X-10 (USA), Inc., 185A LeGrand Avenue, Northvale, NJ 07647
Circle Reader Service Number 234.

PlayWriter Series Expands

Woodbury Computer Associates, Inc.,

has introduced two new titles in its PlayWriter Series of write-your-own-book learning programs: *Mystery!*, a detective book for children nine years of age and older, and *Castles & Creatures*, a fantasy book for children eight and up. With these programs, and the earlier *Tales of Me* and *Adventures in Space* (ages seven to fourteen), children can write, illustrate, print, and bind in hardcover each book they create.

The packages sell for $39.95 each and are available for the Apple II family, Commodore 64/128, and IBM PC/PCjr. Refill packs and teacher's manuals are $9.95 each. Woodbury, in association with Grolier Electronic Publishing, will sponsor a national writing contest this fall with entries handled through schools and retailers.

Woodbury Computer Associates, Inc., 127 White Oak Lane, CN#1001, Old Bridge, NJ 08857

Circle Reader Service Number 235.

IBM, Apple Educational Software

World Book Discovery, Inc., a subsidiary of World Book, Inc., recently released its line of Discovery software for Apple IIe, IIc, and IBM PCjr computers. The series includes 21 programs for children ages three and up.

Discovery software is divided into three categories: Preschool (ages three to five), which focuses on readiness skills like number and pattern recognition); primary (ages six to ten), which offers practice in skills like arithmetic, problem-solving and vocabulary-building; and intermediate (ages ten and up), which helps older students further expand skills learned earlier.

Each series of seven programs is available for $249.95. Individual programs retail for $39.95.

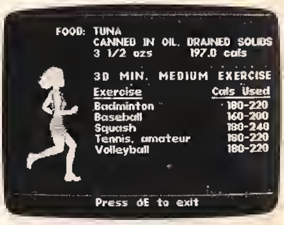
World Book, Inc., The Merchandise Mart, Fifth Floor, Chicago, IL 60654

Circle Reader Service Number 236.

Diet, Adventure Programs

Among several new programs introduced by Bantam Electronic Publishing are *The Complete Scarsdale Medical Diet* ($39.95) for the Apple II series and IBM PC/PCjr, and *The Fourth Protocol*, a graphics and text adventure game based on Frederick Forsyth's bestselling novel, for the Commodore 64/128 ($34.95) and Apple II series ($39.95).

Two adventure programs, the first releases in Bantam's new Choose Your Own Adventure Software Series, are being introduced in September. Entitled *Escape* and *The Cave of Time*, the programs are based on the popular series of books published by Bantam Books, Inc., the software division's owner. They will be available for the Apple II series and for the Commodore 64/128 at a suggested retail price of $34.95.



```
FOOD: TUNA
      CANNED IN OIL, DRAINED SOLIDS
      3 1/2 ozs          197.0 cals

      30 MIN. MEDIUM EXERCISE
      Exercise              Cals Used
      Badminton               180-220
      Baseball                160-200
      Squash                  180-240
      Tennis, amateur         180-220
      Volleyball              180-220

      Press 6E to exit
```

A sample screen from Bantam's The Complete Scarsdale Medical Diet *program for the IBM and Apple computers.*

Bantam has also announced its Micro-Workshop Series of learning software for children. The first three titles in the series are *Fantastic Animals* (ages four through nine), *Creative Contraptions* (ages seven and up), and *Road Rally U.S.A.* (ages ten and up). The emphasis in each package is to encourage creativity while teaching basic learning skills. The IBM PC/PCjr and Apple II-series versions will sell for $39.95, while the Commodore version, to be ready this fall, is set at $34.95.

Bantam Electronic Publishing, 666 Fifth Avenue, New York, NY 10103
Circle Reader Service Number 237.

Fast Apple Disk Drive

The Micro Disk Drive (MDD-640), from Tymac, can store up to four and a half times the information possible on a standard Apple drive and can retrieve information up to 93 percent faster. It can be used with Apple II, II+, and IIe computers. Compatible with both DOS 3.3 and ProDOS, the drive uses 3½-inch disks. Suggested retail price is $399.

Tymac Controls Corporation, 127 Main St., Franklin, NJ 07416
Circle Reader Service Number 238.

New Printer Interfaces

Telesys Computer Peripheral Products has announced several new printer interfaces for Apple, Atari, and Commodore computers. For the Atari, Telesys has introduced the TurboPrint/A ($59.95), a graphics and text parallel printer interface which emulates the printer interface portion of the Atari 850 Interface Module. The TurboPrint/A has external DIP switch access and its own power supply. The TurboPrint/GTA ($99.95) is an advanced graphics and text parallel printer interface with optional plug-in 16K or 32K buffer for Atari computers. It is completely software-compatible with the Atari 850, prints Atari graphics characters (including reverse characters), doubles the printing speed of printers without onboard memory, and has external DIP switches. The B16 16K TurboBuffer ($79.95) and the B32 32K TurboBuffer ($109.95) are available for the TurboPrint/GTA. Both TurboPrint interfaces work with Atari 400, 800, 800XL, 65XE, and 130XE computers.

For the Commodore 64/128 and VIC-20 computers, Telesys has introduced the TurboPrint/C ($49.95), a text-only parallel printer interface; the TurboPrint/GC ($69.95), a parallel interface which prints Commodore graphics including reverse characters, prints four typefaces (normal, expanded, compressed, and expanded-compressed combined), and has external DIP switches; and the TurboPrint/GTC ($89.95), a buffer-expandable parallel interface which prints enhanced Commodore graphics. The TurboBuffers mentioned above are available for the GTC at the same prices.

For the Apple IIe and II+ computers, Telesys has announced the TurboPrint/IIe ($59.95), which prints text with many popular Centronics-type printers and graphics with Epson and Epson-compatible parallel printers. The TurboPrint/IIc ($89.95) performs serial to parallel conversions, has switch-selectable baud rates, and is compatible with most Centronics-type printers. All cables required for installation are included with both interfaces.

Telesys Computer Peripheral Products, 43334 Bryant Street, Fremont, CA 94539
Circle Reader Service Number 239.

Inexpensive Daisy Wheel Printer

Apropos Technology has added a daisy wheel printer to its line of microcomputer printers. The Aprotek Daisy 1120 is equipped with a standard Centronics parallel interface and supports many type fonts, including superscripts, subscripts, underlining, and boldfacing. It has a 2K buffer. Options include an automatic cut sheet feeder ($195) and tractor feed ($82). The printer retails for $364 and has a one-year warranty.

Apropos Technology, 1071-A Avenida Acaso, Camarillo, CA 93010
Circle Reader Service Number 240.

Productivity, Young Learning Packages

Six new educational programs for youngsters ages four through six have been announced by Grolier Electronic Publishing for the Apple II series and the Commodore 64/128 computers at $29.95 per package. Three of the programs—*The Story of Miss Mouse, Rhyme Land,* and *First Steps to Reading: Phonics I and II*—concern reading-readiness. The other three packages—*Exploring Your World: Me and Others, Exploring Your World: The Weather,* and *Play Together, Learn Together*—introduce children to the concepts of body parts, clothing, the weather, and the world around them.

Grolier has also created two new productivity packages, *The Information Connection,* a combination telecommunications program, text editor, and tutorial on one disk for the Apple II family and the IBM PC/PCjr ($59.95 each) and for the Commodore 64/128 ($39.95); and *EduCalc,* a spreadsheet designed to be used in homes and schools, for the Commodore 64/128, Apple II series, and the IBM PC/PCjr ($49.95 home, $59.95 school). The *EduCalc Template,* sold separately for $19.95, features ten application templates preformatted for such home and school applications as budgeting, science, math, and sports.

Grolier Electronic Publishing, 95 Madison Avenue, New York, NY 10016
Circle Reader Service Number 241.

Graphics Control for Commodore

Xetec has introduced the Super Graphix, a graphics interface for Commodore computers. Features include an 8K buffer, ten printing modes, and correct graphics/text aspect ratio for all major printers. Internal fonts support superscripts, subscripts, underlining, boldfacing, and a choice of nine pitches. The Super Graphix comes with a lifetime warranty and retails for $99.95.

Xetec, Inc., 3010 Arnold Rd., Salina, KS 67401
Circle Reader Service Number 242.

More From Mindscape

Mindscape has unveiled several new programs. *The Mist,* based on the Stephen King novella of the same name, and *A View to a Kill,* based on the latest James Bond movie, are text adventures. Each is available for the Apple II line, Apple Macintosh, and IBM PC, and costs $39.95.

Deja Vu is Mindscape's first product developed specifically for the Macintosh. It is a graphics/text adventure in the style of an old 1940s Hollywood mystery movie. It retails for $49.95.

The Luscher Profile, developed in cooperation with Dr. Max Luscher, provides a psychological profile of an individual based on his or her reaction to different colors. It is available for the Apple II line, Macintosh, and IBM PC, for $39.95.

Mindscape, Inc., 3444 Dundee Road, Northbrook, IL 60062
Circle Reader Service Number 243.

Electronic Writing Aids

Simon & Schuster Electronic Publishing Group announced several new titles at the Summer Consumer Electronics Show. Among them is the *Webster's New World Series,* which includes *Webster's New World Spelling Checker* (IBM PC/PCjr, $59.95; Apple II series, $49.95), *Webster's New World Word Processor* (with online thesaurus and spelling checker; IBM PC/PCjr, Apple II series, $124.95), and *Webster's New World Electronic Thesaurus* (IBM PC/PCjr, $59.95).

Simon & Schuster also announced an interactive adventure based on the popular television series *Star Trek. STAR TREK: The Kobayashi Alternative* retails for $39.95, and is available for the IBM PC/PCjr, Apple II series, and Commodore 64.

Simon and Schuster Electronic Publishing Group, Simon & Schuster Building, 1230 Avenue of the Americas, New York, NY 10020
Circle Reader Service Number 244. ©

The Beginners Page

Tom R. Halfhill, Editor

Forget Your Algebra

Don't be misled into thinking that an extensive math background is necessary to program computers. Sometimes, it turns out, too much math knowledge confuses things when you're learning to program.

For instance, the following statement is perfectly acceptable in BASIC, but utter nonsense in mathematics: X = X + 1. It would probably earn you extra homework in a beginning algebra class because one of the first things they teach you is that one side of an equation must equal the other.

But in BASIC, not only is X = X + 1 valid, so is X = X + 2 or even X = X + 10000. Part of the difference is in the way that algebra and BASIC handle the symbol X, called a *variable*. In algebra, a variable is an unknown value; it represents a number you're trying to discover by solving the equation. In BASIC, a variable is a method of storing a value that can change as the program runs. Ordinary numbers are known as *constants*, because numbers don't change. In the statement X = X + 1, the number 1 is a constant, and 1 is always 1.

A variable, on the other hand, is like a flexible number. It can equal anything. And you can change what it equals anywhere in the program. The statement X = 5, called an *assignment statement*, sets the variable X equal to 5. (Actually, X = 5 is an abbreviation for LET X = 5. But the keyword LET is optional in almost all modern versions of BASIC, so it's rarely used anymore.)

After a variable has been assigned the value of 5, the computer treats it like a 5 anytime it subsequently encounters that variable when running the program. The advantage of using a variable instead of a constant to represent 5 is that the variable can be manipulated in a number of ways. Try running this simple program:

```
10 X=5:PRINT X:X=X+1:PRINT X
```

When it's done, you should see the numbers 5 and 6 on the screen, even though the program starts by setting X equal to 5. Why? Because the third statement—X = X + 1—is another assignment statement which adds 1 to the current value of X. Since the current value happens to be 5, then 5 plus 1 equals 6. The final statement prints the new value.

Run the program again after removing the first statement. You'll probably see a 0 and 1 on the screen. That's because almost all personal computers automatically *initialize* variables to zero when the program starts. Be aware, however, that some larger computers don't do this. Instead, the variable may contain an unknown, or *garbage*, value. To keep these garbage values from messing up calculations, programs written for these computers usually begin by initializing all variables to zero.

Variable Names

You're not limited to the letter X as a variable name, of course. You can use any letter from A to Z. Longer names are possible, too, and help make your programs easier for others (and even yourself) to understand. For instance, if you need a variable to hold the sum of a series of numbers added together, SUM is more readable than S.

Different versions of BASIC have different rules for variable names. In Commodore and Applesoft BASIC, variables can consist of letters and numbers but no symbols, as long as the first character is a letter. A1 is allowed, but not 1A. Commodore and Apple variables can be of any length, but only the first two characters are *significant*. That means the computer looks only at the first two characters of the name to decide if it's unique. SUM and SAM are treated as different variables, but SUM1 and SUM2 are not. Watch out for this, because it can lead to mysterious programming bugs.

Also, Commodore and Applesoft BASIC (and most other versions of BASIC) don't allow variables with *reserved words*. That is, any word that BASIC recognizes as a command, statement, or function cannot be part of a variable name. This restriction, too; can lead to mysterious errors. An example is the variable TOTAL. It looks as innocent as SUM, but contains the keyword TO (which is part of the FOR/NEXT loop statement, as in FOR X = 1 TO 10).

IBM BASIC permits variables with letters, numbers, and decimal points, as long as the name starts with a letter. Names can be of any length, and the first 40 characters are significant. Although a variable cannot be a reserved word, it can *contain* a reserved word. Therefore, the variable TOTAL is okay but the variable TO is not.

In Atari BASIC, variables may contain letters and numbers, as long as they start with a letter, and can be of any length with *all* characters significant. What's more, variables can include reserved words or even consist of a reserved word if the assignment statements use the optional keyword LET. Thus you can have a statement such as LET LET = LET + LET. In TI BASIC, variables are limited to 15 characters (all significant) and can start with either a letter or one of the following symbols: @, [,], /, and _. Oddly, though, the rest of the name cannot *contain* a [,], or /.

Up to now we've been discussing *numeric* variables—variables that represent ordinary numbers. Next month we'll examine other types of variables.

Computers and Society

David D. Thornburg, Associate Editor

Compilers, Interpreters, And Flow: Conclusion

Over the past two columns I've explored some ways in which programming with an interpreter or compiler can influence the nature and complexity of the programs we write. As this is written, I'm approaching the end of a Logo-based programming course that I've been teaching to graduate students at Stanford. (Yes, Virginia, there is Logo after second grade!) Because I wanted my students to have access to a high-speed runtime language, I elected to use a Logo compiler in this course.

As was mentioned last month, the speed improvements in compiled programs have a lot to do with the program's ability to maintain a sense of "flow" with the user. But, just as the compiler's benefits are directed toward the user, interpreters provide quite a few benefits to the programmer—especially if the programmer is just learning to use the language. When computer languages are taught in school, the assignments and lectures usually structure the learning process for the students, and the work at the keyboard tends to reinforce what has already been learned rather than encourage new discoveries. It is when learning a new language on your own that an interpreter is of tremendous value.

Instead of studying a new language in a book before trying to create programs, I usually jump in with both feet and start sloshing around, trying to get something to work. In educational circles, this experimental learning style is called *discovery-based learning*. In the realm of videogames, people like Bernie DeKoven call it "learning by dying." One of the reasons videogames can be learned without referring to extensive manuals is that you can usually figure out what caused you to lose your turn or one of your "lives," so you can avoid that mistake the next time.

A well-designed interpreter and program editor could allow people to master new programming languages in this way. (This approach could also be applied to education in general, but that's a topic for another column.)

Bug Detectors

One example of this is Macintosh Pascal. Mac Pascal contains both an interpreter and a powerful program editor that allows beginners to learn this language in a highly interactive and self-paced fashion. Those of you who know Pascal may think that the "sloshing around" style of learning is ill-suited to a language whose structure is more like a faceted jewel than a lump of clay. But I believe the rigid structure imposed on Pascal programs makes an "intelligent" editor and program interpreter of tremendous value.

The program editor automatically indents program lines and boldfaces Pascal keywords, making the listing very easy to scan. Furthermore, if the interpreter detects an error as the program is running, helpful "bug detection" tools point out the line with the problem and provide as much help in fixing the problem as possible.

This interaction between the interpreter and program editor encourages the programmer to try new constructs and ideas, safe in the knowledge that "bad grammar" will be detected and clearly identified.

The interaction between the interpreter and program editor does not stop here. You can also execute programs line by line, place "stop signs" at various locations in the program to help debug the code, and even create windows to show the values of certain variables as the program runs.

Normally, Pascal doesn't allow you to execute single-line programs. But Macintosh Pascal does, so you can type fragments of Pascal code to see how they behave. This makes the language far easier to learn. Fortunately, Mac Pascal is being adapted for the Apple IIe and IIc computers as well, thus bringing this style of Pascal programming to a far larger audience.

The Best Compromise

The choice between an interpreter or a compiler, then, depends on the application and the point of view. From the user's perspective, compiled programs have the advantage of execution speed. For programmers, interpreters have more advantages. Since most programs involve both users and programmers, this suggests that widely used programming languages should be available in two forms—an interpreter for creating and testing programs, and a compiler to produce the final product.

Furthermore, it's essential that these modules be compatible with each other's source code. Programmers should be able to take a program that was written and debugged with the interpreter and drop it into the compiler to generate the highly efficient runtime code for the user.

As progress continues along these lines, we'll see a trend toward application programming in increasingly higher-level languages. No longer will programmers have to learn machine language to build industrial-strength programs. Anyone who knows how to write in high-level languages will be able to create efficient programs of all types for their own use, as well as for the use of others.

David Thornburg welcomes letters from readers, but regrets that he cannot personally answer all his mail. Correspondence should be sent in care of COMPUTE!. ©

Telecomputing Today

Arlan R. Levitan

SIG Wars

You may recall that last month we raised the question of what the commercial information services would do about system operators (sysops) of special interest groups (SIGs) or discussion forums who were beginning to set up branches of their SIGs on competing services.

The shoe has finally dropped. In May, users of the Delphi information service noticed that the Delphi branch of MAUG (Micronetworked Apple User Group) mysteriously vanished after a couple of weeks of existence, to be replaced by a generically named Apple SIG with a new sysop.

Apparently CompuServe, the current SIG heavyweight among information services, was still smarting from the wholesale defection of its Commodore forum sysops to another competing service. In any case, CompuServe won back the sysop of MAUG (its most popular SIG forum) with an offer that couldn't be refused.

Shortly after the disappearance of MAUG/Delphi, MAUG/CompuServe became *three* SIGs: one for Apple II owners, a second for Macintosh fans, and a third for Apple software and hardware developers. All of the SIGs remained under the able tutelage of the original MAUG sysop, who ended up with three SIGs rather than one (or zero).

This incident does raise some disturbing issues which should be aired and discussed within the telecomputing community. At the conclusion of this column, I'll give you a way to participate in this debate.

Two Points Of View

A lot of users cried foul after the MAUG affair, accusing one of the parties involved of restraint of trade and illegal chicanery. Much of this was mildly sour grapes from MAUG regulars who had regarded MAUG/Delphi as welcome relief for their pocketbooks. MAUG/Delphi's off-shift hourly rate for 1200 bits-per-second (bps) modems was half that of CompuServe's. In fact, Delphi's off-shift rate even for 2400 bps was still less than CompuServe's 1200 bps charges. (CompuServe is the leading information service, so its competitors are offering lower rates in an effort to entice customers.)

Setting emotions aside for a minute, there is no evidence that anyone involved in the MAUG incident abrogated the legal rights of any other party. As for whether the negotiations tended toward "hard ball," all I can do is remind mild-mannered telecomputerists that in the words of Jack Tramiel, "business is war."

Users who regularly upload public domain software to SIGs get little in return other than bills for their connect time. Shouldn't there be a greater reward than simply a pat on the back?

The situation does have aspects of David versus Goliath though, and since we love to root for the underdog (even when Sweet Polly isn't involved), it's hard on a gut level not to side with the sysops. Even the most influential sysops tend to have less bargaining power than corporations with legal staffs.

Who Owns The Info?

Another issue that tends to bother many telecomputing regulars is the question of who owns (or who they think should own) the information contained in a SIG. By the terms of most information service user contracts, the contents of both the message base and program download areas are the property of the service. Yet, the messages and the files uploaded to the program area are provided by the users. So SIG users pay the information service to distribute their messages and programs.

There is little doubt that a case may be made for the information service owning the message base, but what about ownership of the public domain programs?

Users who regularly upload public domain software to SIGs get little in return other than bills for their connect time. Shouldn't there be a greater reward than simply a pat on the back? Many noncommercial bulletin board systems offer special benefits to regular contributors. Why shouldn't commercial services do the same?

To be perfectly fair, SIG users do receive value from the service in the form of replies to messages and software to download. Hopefully the value received is commensurate with the tariffs levied.

Time For An E-Poll

How do you feel about this issue? Am I being too tough or not tough enough on the information services? Am I off base or stealing home on a suicide squeeze? E-mail your opinions to me and I'll print the results of our electronic mini-poll in the months to come.

Arlan R. Levitan
Source ID: TCT987
Delphi: ARLANL
People Link: ARLANL
CompuServe: 70675,463

©

The World Inside the Computer

Fred D'Ignazio, Associate Editor

A Robot Toddler

A couple of months ago, the Heath Company of Benton Harbor, Michigan sent me a HEROjr personal robot to review on the PBS show *The New Tech Times*. HEROjr costs $600 in kit form and is a 19-inch tall, 22-pound comedian. He comes with a repertoire of slapstick sayings (like "Nanu! Nanu!" and "Beam me up, Scotty!"), corny songs (like "Old MacDonald Had a Robot"), and special robot games (like "Cowboys and Robots"). He can order a hamburger and fries at MacDonald's, imitate a Dr. Pepper commercial, and carry on an animated conversation with a vacuum cleaner that he has mistaken for a human being.

Despite his impressive technical credentials—including full programmability, speech output, light, sound, and infrared sensors, ultrasonic sonar, a clock/calendar, a burglar alarm, a 17-key keypad, an RS-232 interface, and whatnot—HEROjr has an aura of lovable vulnerability. He is not very tall, he talks in a shy little voice, and he is single-minded about looking for human beings to play with or serenade. If he were a little smaller, he'd make a perfect lap robot.

During the day, HEROjr wanders around our house singing, gabbing, and reciting nursery rhymes. He is about the size of a toddler and he acts like a toddler. He is unpredictable, has a mind of his own, and frequently gets into mischief. I keep a toddler gate at the top of the stairs, since most of HEROjr's exploring takes place on the second floor of our house, and I wouldn't want him falling down the steps.

The main difference between HEROjr and a toddler is that when you want HEROjr to take a nap, you just push the SLEEP switch on the back of his head. This feature comes in handy when HEROjr gets himself stuck under the kitchen table, or when you want to plug a new personality cartridge into his brain. Or when his two six-volt, nickel-cadmium batteries are low and you need to recharge them.

HEROjr got a chance to see something of the world recently when I received a speaking invitation from the School Trustees Association in Vancouver, British Columbia. The school trustees (equivalent to school board members in the U.S.) were having their annual meeting, and they wanted me to speak about the future of computers in schools. I had become so attached to HEROjr by this time that at the last minute I decided to take him along.

There's A Robot On This Airplane!

Our trip began with HEROjr riding with me in the back of a taxicab to the Roanoke airport early one morning to catch a plane to Chicago. When I introduced the robot to Red Eye, my favorite Roanoke cabbie, Red Eye said, "Junior, eh? That's a good name for a robot!"

From that point on, HEROjr became "Junior."

Junior and I spent the rest of that day catching planes and running frantically across airports trying to make connecting flights. People reacted to Junior in a variety of ways. A few were hostile—like the flight attendant on one airline who wouldn't say hi to Junior "Because," she said (obviously having given great thought to the matter), "I don't say hi to robots!" But most people were openly curious and receptive. And some had a strong tendency to anthropomorphize the robot. They wanted to talk with Junior, play with him, protect him, and care for him. For example, one flight attendant wasn't comfortable until she had tucked a pillow behind Junior's head and a blanket around his wheels—"Just in case he gets chilly," she explained with a smile.

On the plane from Chicago to Seattle, I overheard a woman in the seat ahead of me asking her husband about Junior. "I hope the robot has its seatbelt on," she said.

But Junior wasn't wearing his seatbelt. He was sleeping in the coat closet at the back of the airplane because it was the only place he would fit, and also because it kept him hidden from nervous passengers and unfriendly flight attendants. Suddenly our plane hit some turbulent weather, and Junior apparently bumped into a hanging bag hard enough to throw his switch from SLEEP to NORM. Instantly Junior woke up and began singing to someone's overcoat. "Daisy, Daisy," he crooned, "Give me your answer, true. I'm half crazy, all for the love of you...."

The passengers near the coat closet began laughing, but some passengers were worried, too. "Who is that in there?" asked one man. Another cried, "There's a robot on this airplane!"

The flight attendant rushed to my seat in the forward section of the plane and took me to Junior's rescue. By the time I got there, he was screaming "Help! Help! Help!" This means that he had tried to explore but couldn't, because his wheels were stuck. As I reached into the coat closet and pushed his switch back to SLEEP, the flight attendant said, "I tried to calm him by telling him that you were coming. But he just kept crying for help."

Next month I'll tell you some more of Junior's adventures, and I'll have some thoughts about how people react when they meet their first real robot—up close and in person. ©

Donald B. Trivette

The Mysterious Editors

Recently I asked a group of computer users—mostly those with IBM PCs—how many used an editor. I got a blank stare. Most had only the vaguest idea of what an editor is and what you do with one—the consensus being that editors are either useless or redundant. (Self-preservation prevents me from making a comparison between the software and the profession.) No one confessed to actually owning an editor, yet everyone who has an IBM PC or PCjr has at least three of them.

An editor is a program that allows you to enter text, numbers, or other data (binary, hexadecimal, etc.) into the computer's memory; to display, modify, and change that data; and to store and retrieve it using an external device such as a disk drive. You may recognize that word-processing programs fall within this definition, for word processors are in fact very fancy editors. Most of the commands (and complications) of a word processor are for formatting and printing text in a *pretty* way—the actual editing commands are relatively few and easy to use.

The first editor IBM gives you is built into the hardware. It's a part of the BASIC language—the part that allows you to type BASIC statements and to move the cursor around the screen with the arrow keys. This is called *full-screen editing*. The BASIC editor comes up automatically when you turn on a PC or PCjr without a disk in the drive, or when you type BASIC (or BASICA) at the DOS A> prompt (the PCjr requires Cartridge BASIC in this case). It's a special-purpose editor designed to make entering and correcting BASIC statements easy, and it can't really be used for anything else. Nevertheless, it *is* an editor.

The second editor IBM gives its users is on the DOS disk and is named DEBUG. This is also a special-purpose editor. Using DEBUG, a programmer can follow the step-by-step execution of a machine language program and trace the contents of memory as it changes. DEBUG can also be used to display and change the contents of a file—particularly a program file containing machine language instructions. However, you must know something about machine language to use DEBUG effectively.

The third editor is one almost no one uses, although it too comes on the DOS disk. It's called EDLIN for LINe EDitor. The story goes that some programmers at Microsoft put together a quick and dirty editor for their own use while working on the then-secret IBM PC project. When IBM bought DOS and BASIC from Microsoft, the editor was shipped along by mistake. Supposedly some folks at IBM thought EDLIN was supposed to be a consumer product, so it was included on the DOS disk along with BASIC and DEBUG. What was intended to be an internal tool has now permeated thousands of homes and offices.

The Ugly Duckling

Neither Microsoft nor IBM is especially proud of EDLIN. It doesn't showcase the PC's power, so it remains the ugly duckling of IBM software: Still, it has many of the requisites for a general-purpose editor: You can use it to create, display, and modify a file, and you can use it to save and load files. If only it had a print command, it might have been the PC's first word processor. And if it supported full-screen editing like BASIC, instead of primitive line-editing, it might be one of the PC's most popular programs. Still, it's not a totally useless editor—once you get used to it.

Some rainy Saturday, when you want to learn something new, take out your DOS disk and try EDLIN. The documentation is in the DOS manual, and you're likely to need it. Here are a few tips:

- At the A> prompt, type EDLIN *and* the name of the file you want to edit. EDLIN won't start unless you give it the name of a file, new or existing, when you start the program.
- The DOS disk is write-protected, so either copy EDLIN to another disk or edit a file on drive B. For example, to edit a new file named ABC on the disk in drive B:, type EDLIN B:ABC.
- The asterisk (*) you'll see when EDLIN is active is the EDLIN prompt, just as A> prompts for DOS and Ok for BASIC.
- EDLIN comes up with the * prompt. To begin entering input, type an I (for input mode) at the prompt.
- Line numbers are typed *before* editor commands. For example, to list lines 20 through 30, the command is 20,30L. This is exactly backward from BASIC.

There *are* some reasons, other than curiosity, to use EDLIN. It has so few commands (14) that it's super compact. The whole program is just 4600 bytes long. That means there's room for EDLIN on almost any disk, so you can always have an editor online to create a new BATCH file or even to quickly modify a text file. And because it's so small, there's lots of memory left for the file itself—an important consideration for PCjr users. More than once on the Junior I've had to use EDLIN to edit a file too large for my memory-hungry word processor. That's when an ugly duckling truly becomes a swan.

Donald B. Trivette is the author of Putting Jr to Work: A Guide to the IBM PCjr, *published by COMPUTE! Books.*©

Programming the TI

C. Regena

The OPEN Statement

Recently I received a call from a young programmer who wanted to know more about the OPEN statement. I really couldn't give him an adequate answer over the phone ("look at your manuals"), so I'll give several examples here.

The OPEN statement means about the same thing in all versions of BASIC, but each computer has its own variations. As the statement implies, the function of OPEN is to open a *file*—or, as I like to think of it, to get the attention of another device to be used with the main console. Various forms of the OPEN statement are described in the manuals that come with the peripherals.

OPEN statements are generally followed by the number of the device you want to address. In TI BASIC, you may use any constant or variable with a value of 1 to 255 for the device number. The number is preceded by the # sign, such as OPEN #1: to open file #1.

Whenever you use an OPEN statement, it is good programming practice to include a CLOSE statement when you're finished with the device. If your program stops with an error, the files are automatically closed.

Speech Synthesis

If you have the TI Speech Synthesizer and the *Terminal Emulator II* command module, use an OPEN statement to make the computer talk:

```
OPEN #1:"SPEECH,"OUTPUT
```

This alerts the speech device to be ready for output. Then all you need is a PRINT #1 statement (pronounced "print file one"):

```
PRINT #1:"HELLO"
```

Within a program, you can print on the screen with a regular PRINT statement and produce speech with the PRINT # statement:

```
10 OPEN #5:"SPEECH,"OUTPUT
20 PRINT "THIS IS A TEST."
30 PRINT #5:"THIS IS A TEST."
40 CLOSE #5
```

By the way, if you'd like to hear your program listing, use the command LIST "SPEECH."

Printing

To get the most out of a printer, you really need to study your printer and interface manuals. The Texas Instruments RS-232 interface manual shows all the different parameters for accessing your printer. Here are some examples of OPEN statements:

```
OPEN #1:"TP"
OPEN #1:"PIO"
OPEN #1:"RS232.BA=600"
OPEN #1:"RS232.TW.BA=110"
```

Once you've determined the necessary OPEN statement for your hardware configuration, you can use PRINT #1 (or whatever file number you opened) to send any command to the printer. If someone else wants to modify your program for another configuration, they can simply change the OPEN statement for their setup.

PRINT # lets you print constants, variables, and strings. You can align columns with the TAB function. In Extended BASIC, the PRINT #1, USING statement also is handy to format the output. Here's a short example of sending output to the printer:

```
10 OPEN #1:"RS232.BA=600"
20 PRINT #1:TAB(10);"THIS SHOULD
     PRINT."
30 CLOSE #1
```

File Processing

If you want to learn more about file processing with the OPEN statement, the manual that comes with the TI-99/4A contains a good description of various forms of OPEN. I also discussed file processing in my COMPUTE! columns of March, April, and May 1984. And a program which saves names and addresses on cassette is in my book, *Programmer's Reference Guide to the TI-99/4A*.

This month's example program shows how to use the OPEN statement to save a drawing on cassette. Type in and run the program, then press the arrow keys to draw a low-resolution picture on the screen. When you're done, press CTRL-S to save the picture on tape. You can load it by pressing CTRL-L.

The program uses different character numbers for the different-colored drawing squares. These are defined in lines 140–200. When the program loads a picture, it uses the character numbers to determine the locations of the colored squares.

Lines 540–870 contain the drawing procedure. The variable X is the row and Y is the column. C is the character number. If you press the space bar, C is incremented by 4 and the color of the square changes. The arrow keys move the square, and it stops at each screen edge.

Lines 890–990 keep track of the character numbers for each column in each row if you want to save the picture. Lines 1000–1050 save the strings of G$, which contain the character numbers on cassette. The procedure takes quite a while because each item saved has its own leader. You can hear the cassette recording during this process. The OPEN statement in line 1000 opens device #1 as "CS1," or cassette, for OUTPUT. INTERNAL and FIXED are two options available in the OPEN statement for cassette that specify how to save the data. FIXED 96 is used because each G$ will be 96 characters long.

Lines 1150–1210 load the picture from cassette. Notice how the OPEN statement in line 1160 matches the format of line 1000, except that it specifies INPUT instead of OUTPUT. The INPUT #2 statement reads G$ row by row.

Input variables must match the way they were previously saved, although you can use different variable names. Lines 1230–1320 recreate the picture on the screen from the information read off tape.

If you'd like to save typing effort, you can obtain a copy of this program by sending a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Doodle With CS1

```
100 REM  DOODLE WITH CS1
110 DIM G$(24)
120 CALL CLEAR
130 PRINT TAB(11);"DOODLE":
    :::
140 FOR C=10 TO 16
150 D=C*8+24
160 CALL CHAR(D,"")
170 CALL CHAR(D+4,"FFFFFFFF
    FFFFFFFF")
180 CALL COLOR(C,C,C-7)
190 NEXT C
200 CALL COLOR(10,2,3)
210 PRINT "CHOOSE:"
220 PRINT :"1 DRAW"
230 PRINT :"2 LOAD PICTURE"
    :::
240 CALL KEY(0,K,S)
250 IF K=50 THEN 1160
260 IF K<>49 THEN 240
270 REM
280 CALL CLEAR
290 PRINT "PRESS SPACE BAR
    TO CHANGE"
300 PRINT "SCREEN COLOR."
310 PRINT :"PRESS <ENTER> F
    OR DESIRED{3 SPACES}COL
    OR."
320 SC=3
330 CALL SCREEN(SC)
340 CALL SOUND(100,1497,2)
350 CALL KEY(0,K,S)
360 IF K=13 THEN 420
370 IF K<>32 THEN 350
380 SC=SC+1
390 IF SC=10 THEN 380
400 IF SC=17 THEN 320 ELSE
    330
410 REM
420 CALL CLEAR
430 PRINT "MOVE ARROW KEYS
    TO DRAW."
440 PRINT :"PRESS SPACE BAR
     TO CHANGE{3 SPACES}COL
    ORS."
450 PRINT :"PRESS CTRL S TO
     SAVE."
460 PRINT :"PRESS CTRL L TO
     LOAD."
470 PRINT :"PRESS CTRL E TO
     END."
480 PRINT ::"NOW PRESS ANY
    KEY TO START."
490 X=12
500 Y=16
510 C=104
520 CALL KEY(0,K,S)
530 IF S<1 THEN 520
540 REM  DRAW
550 CALL CLEAR
560 CALL SCREEN(SC)
570 CALL KEY(0,K,S)
580 CALL HCHAR(X,Y,32)
590 CALL HCHAR(X,Y,C)
600 IF K=147 THEN 890
610 IF K=140 THEN 1160
620 IF K=133 THEN 1350
630 IF K<>32 THEN 680
640 C=C+4
650 IF C<>160 THEN 570
660 C=104
670 GOTO 570
680 IF K<>69 THEN 730
690 X=X-1
700 IF X>0 THEN 570
710 X=1
720 GOTO 570
730 IF K<>83 THEN 780
740 Y=Y-1
750 IF Y>0 THEN 570
760 Y=1
770 GOTO 570
780 IF K<>68 THEN 830
790 Y=Y+1
800 IF Y<33 THEN 570
810 Y=32
820 GOTO 570
830 IF K<>88 THEN 570
840 X=X+1
850 IF X<24 THEN 570
860 X=24
870 GOTO 570
880 REM  SAVE
890 CALL SOUND(150,1200,2)
900 FOR ROW=1 TO 24
910 G$(ROW)=""
920 FOR COL=1 TO 32
930 CALL GCHAR(ROW,COL,G)
940 IF G<>32 THEN 960
950 G=200
960 G$(ROW)=G$(ROW)&STR$(G)
970 NEXT COL
980 CALL SOUND(50,1200,2)
990 NEXT ROW
1000 OPEN #1:"CS1",OUTPUT,I
     NTERNAL,FIXED 96
1010 FOR ROW=1 TO 24
1020 PRINT #1:G$(ROW)
1030 NEXT ROW
1040 PRINT #1:X,Y,C,SC
1050 CLOSE #1
1060 PRINT ::"CHOOSE:"
1070 PRINT :"1 GO BACK TO S
     AME DRAWING"
1080 PRINT :"2 START NEW DR
     AWING"
1090 PRINT :"3 SAVE ANOTHER
      COPY"
1100 PRINT :"4 LOAD PICTURE
     "
1110 PRINT :"5 END"
1120 CALL KEY(0,K,S)
1130 IF (K<49)+(K>53)THEN 1
     120
1140 ON K-48 GOTO 1230,280,
     1000,1160,1350
1150 REM  LOAD
1160 OPEN #2:"CS1",INPUT ,I
     NTERNAL,FIXED 96
1170 FOR ROW=1 TO 24
1180 INPUT #2:G$(ROW)
1190 NEXT ROW
1200 INPUT #2:X,Y,C,SC
1210 CLOSE #2
1220 REM
1230 CALL CLEAR
1240 CALL SCREEN(SC)
1250 FOR ROW=1 TO 24
1260 FOR COL=1 TO 32
1270 G=VAL(SEG$(G$(ROW),COL
     *3-2,3))
1280 IF G<>200 THEN 1300
1290 G=32
1300 CALL HCHAR(ROW,COL,G)
1310 NEXT COL
1320 NEXT ROW
1330 GOTO 570
1340 REM
1350 CALL CLEAR
1360 END
```

©

INSIGHT: Atari

Bill Wilkinson

Using Serial Input/Output

Last month, I introduced the structure of Atari's operating system (OS). My most important point was that the OS consists of several layers. When you type in a BASIC statement such as LPRINT "Hi There!", you cause a fairly complex chain of events. First, BASIC figures out that LPRINT means you want to use a printer, so it calls the OS to open a channel to the printer (always channel number 7, in this case). Then BASIC sends the bytes to be printed to a part of the OS called Central Input/Output (CIO), which in turn realizes that a file to the printer has been opened on that channel. CIO calls the printer driver, which collects bytes until it has a block of them (or until it gets a carriage-return character or a CLOSE command). Finally, the printer driver sends a block of bytes to the printer by calling *Serial Input/Output* (SIO)—another subroutine inside the OS, and the subject of this month's discussion.

I'd like to point out that this process stops at SIO only as far as the computer is concerned. The printer interface (for example, an 850 Interface Module) also contains a microprocessor which collects the block sent to it by SIO. Then the interface passes the block, a byte at a time, to the printer. Within the printer, yet another microprocessor is usually employed to control the various motors and hammers and wheels that actually place the characters on paper.

Did you note that the process of printing even a single character most probably requires the use of three microprocessors? Did you stop to think that each of these processors requires software to make it work? Did you ever wonder why there are so many people making a living at programming? (Though barely, in the case of some of us.)

Perhaps the most amazing thing is that, for the most part, the three microprocessors work reliably and efficiently together. (It is even more amazing when you consider that either the printer or interface module is often made by a company other than the one which made the computer!) The secret to success here is standardization. The usual printer connection is a fairly simple one, originally defined by a company named Centronics and now adopted by almost every manufacturer in the microcomputer market.

The way your Atari computer "talks" to your interface module, though, is strictly an Atari invention—the SIO. There is a well-defined protocol associated with SIO. It includes such niceties as Command and Data Frames, Acknowledgment, Nonacknowledgment, Command and Bus Errors, and more. Luckily, 99 percent of all Atari programmers need never learn these gory details, since there really isn't anything you can do to change their workings.

Disk Access Via SIO

Some programmers, however, do want to send and receive blocks via SIO. And usually the blocks to be transferred are disk sectors. So let's look at how one reads or writes a specific disk sector.

When SIO is called by a program, it expects to find certain information in a *Device Control Block* (DCB). There is only one DCB, located at $0300–$030B (768–779 decimal). It contains four one-byte values and four two-byte (word) values, all of which must be set up properly. The accompanying table briefly describes each location in the DCB. See COMPUTE! Books' *Mapping the Atari* for more details.

Does all this look confusing? Not to worry. Program 1 below is a subroutine which does most of the work for you. Just type it in, LIST it to disk or cassette, and use it in your own programs whenever you wish. Program 2 demonstrates how to use the subroutine, though I hope the comments make it pretty much self-explanatory. (Perhaps I should note that a command of R reads a sector, P writes a sector without verifying it, and W both writes and verifies a sector.) To use Program 2, you must add the subroutine from Program 1. You can either type in the lines from Program 1, or ENTER them from disk or tape if you have LISTed out a copy of Program 1. Program 3 is the source code behind the DATA statements in line 9210 of Program 1.

If you type in and use Program 2, you might like to remember that the *volume table of contents* (VTOC) of a DOS 2.0-compatible disk is in sector 360. The directory occupies sectors 361 to 368. Sectors 1, 2, and 3 are for booting only. All other sectors from 4 to 719 should be DOS file sectors. (See COMPUTE! Books' *Inside Atari DOS* for more info. Caution: The diagram of the sector link bytes is wrong.)

Finally, I give you a hint and challenge for next month: Most drives not made by Atari allow the user to specify their configuration (for example, single or double density). You can read their configuration blocks with an SIO command of N (or write via O). But be careful! DSIZE must be given as 12 bytes. Can you modify our subroutine to read the configuration block? Good luck.

DCB Layout Table

Location		Name	Size	Purpose
Hex	Dec			
300	768	DDEVIC	1	Name of device on SIO bus (all disk drives use "1," $31, as a name).
301	769	DUNIT	1	Unit number of device (to distinguish D1: from D2:, for example).
302	770	DCOMND	1	Command, usually an ATASCII letter, such as "R" for read sector (but "!" will format a disk!).
303	771	DSTATS	1	Direction control before call to SIO; status of operation upon return.
304	772	DBUF	2	Address of buffer to read from or write to, as appropriate.
306	774	DTIME	2	Timeout value. SIO waits this many seconds before giving up.
308	776	DBYTE	2	Number of bytes to transfer (always 128 or 256 for disks).
30A	778	DAUX	2	Purpose varies; always sector number when used with disks.

Program 1: SIO Subroutine

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
LF 9000 REM ....................
        ..............
JG 9010 REM DISK SECTDR I/D
        RDUTINE
JF 9020 REM . ENTER:
JD 9030 REM .{3 SPACES}secto
        r number in SECTOR
ND 9040 REM .{3 SPACES}drive
         number in DRIVE
DC 9050 REM .{3 SPACES}buffe
        r address in ADDR
IP 9060 REM .{3 SPACES}comma
        nd in CMD$
NJ 9070 REM .{3 SPACES}densi
        ty in DENSITY
GM 90B0 REM (only "R","W","P
        " are valid for CMO$
        )
EA 9090 REM (only 1=SGL and
        2=DBL are valid for
        DENS1TY)
FA 9100 REM . EXIT:
CH 9110 REM .{3 SPACES}statu
        s in SIDSTATUS
LA 9120 REM
OI 9160 TRAP 9220:REM activa
        ted if SIDCALL$ alre
        ady DIM'd
IO 9170 DIM SIDCALL$(16)
MC 91B0 RESTORE 9210
JP 9190 FOR CNT=1 TO 14:READ
         BYTE
EN 9200 S1OCALL$(CNT)=CHR$(B
        YTE):NEXT CNT
NC 9210 DATA 104,32,B9,22B,1
        73,3,3,133,212,169,0
        ,133,213,96
FB 9220 TRAP 40000:REM turn
        off TRAP
NO 9230 PDKE 76B,ASC("1"):RE
        M don't ask me why
GC 9240 PDKE 769,OR1VE:REM m
        ust be 1 through B
OJ 9250 PDKE 770,ASC(CMD$)
DN 9260 POKE 771,12B:REM ass
        ume write
LP 9270 1F CMO$="R" THEN POK
        E 771,64
HA 92B0 POKE 773,1NT(ADDR/25
        6):REM buffer addres
        s
PF 9290 PDKE 772,AOOR-256*PE
        EK(773)
FB 9300 POKE 774,3:REM short
         timeout
JK 9310 PDKE 775,0:REM (high
         byte of timeout)
AA 9320 POKE 776,12B:POKE 77
        7,0:REM assume singl
        e density
LG 9330 IF DENS1TY=2 THEN PO
        KE 776,0:PDKE 777,1
YK 9340 PDKE 779,1NT(SECTOR/
        256)
LD 9350 PDKE 77B,SECTOR-256*
        PEEK(779)
HN 9360 SIDSTATUS=USR(AOR(SI
        DCALL$))
LD 9370 RETURN
```

Program 2: SIO Demo

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
KC 1000 REM PRDGRAM TD DEMDN
        STRATE SECTOR READ S
        UBROUTINE
HJ 1010 REM NDTE: rather tha
        n ask questions, we
EB 1020 REM .{5 SPACES}assum
        e that we will work
        with drive
KP 1030 REM .{5 SPACES}numbe
        r 1 and that it is s
        ingle
HK 1040 REM .{5 SPACES}densi
        ty (12B byte sectors
        )
KK 1050 REM
PA 1100 DIM BUFFER$(256):REM
         guaranteed adequate
ML 1110 ADDR=ADR(BUFFER$):RE
        M required by subrou
        tine
PI 1120 DR1VE=1:REM assumpti
        on...easily changed
HC 1130 DENSITY=1:REM assump
        tion...ditto
JO 1140 DIM CMD$(1):CMD$="R"
        :REM always, for thi
        s demo
KL 1150 REM
NB 1160 PRINT "What sector t
        o display";
CJ 1170 1NPUT SECTOR
BD 11B0 GOSUB 9000
EM 1190 GRAPHICS 0
DL 1200 PRINT "Read Sector "
        ;SECTDR;" gave Statu
        s ";SIDSTATUS
OP 1210 SIZE=DENSITY*12B:REM
         size is 12B or 256
CJ 1220 SECTDR=PEEK(ADDR+SIZ
        E-3)
JC 1230 FILE=INT(SECTDR/4)
EP 1240 SECTDR=SECTOR-4*FILE
ON 1250 SECTDR=SECTOR*256+PE
        EK(ADDR+SIZE-2)
EA 1260 CNT=PEEK(AODR+SIZE-1
        )
DO 1270 PRINT "If DDS file s
        ector, this is file
        #";FILE
NB 12B0 PRINT "  there are "
        ;CNT;" bytes in this
         sector"
NA 1290 PRINT "  and the nex
        t sector is number "
        ;SECTOR
FB 1300 PRINT
JL 1310 FDR LINE=0 TD DENSIT
        Y*12B-1 STEP B
FP 1320 BYTE=L1NE:GDSUB 1500
        :PR1NT ":";
NK 1330 FDR CNT=0 TO 7
PD 1340 BYTE=PEEK(ADOR+LINE+
        CNT):GDSUB 1500:PRIN
        T " ";
DN 1350 NEXT CNT
NN 1360 FOR CNT=0 TD 7
DA 1370 BYTE=PEEK(ADDR+LINE+
        CNT)
AD 13B0 IF BYTE>127 THEN BYT
        E=BYTE-12B
BB 1390 PRINT CHR$(27);CHR$(
        BYTE);
DJ 1400 NEXT CNT
FD 1410 PRINT
CO 1420 NEXT L1NE
FF 1430 PR1NT
NK 1440 GOTO 1160
LA 1450 REM ....................
        ....................
        ...
PF 1460 REM A QUICKY DECIMAL
         TD HEX CDNVERTER
NF 1500 TRAP 1520
DO 1510 DIM HX$(16):HX$="012
        34567B9ABCDEF"
PD 1520 TRAP 40000
EK 1530 HX=INT(BYTE/16)+1:PR
        INT HX$(HX,HX);:HX=B
        YTE-16*HX+17:PRINT H
        X$(HX,HX);
KK 1540 RETURN
```

Program 3: Subroutine Source Code

Note: This listing is provided for informational purposes; it requires an assembler to enter into your computer.

```
    *= anyplace
CALLSIO
  PLA      ;throw away count
           ; of arguments
  JSR SIOV   ;(at $E459)
  LDA DSTATS ;SIO status
             ; (from DCB)
  STA FR0 ;floating point
           ; register 0, $D4
  LDA #0
  STA FR0+1 ;(to get a two-
            ; byte value)
  RTS ;back to BASIC caller    ©
```

Jump Search

Jerry Sturdivant

Learn how the binary search method can speed up data handling. The short demonstration program listed below runs on the Atari 400/800, XL, and XE series; Apple II-series; IBM PC/PCjr; all Commodore computers; TI-99/4A; the Radio Shack Color Computer; and other personal computers with BASIC.

Searching for a specific item in a collection of data is a fundamental computing task. Word processors, databases, and address book programs all need to locate data quickly and accurately. This article shows how to use the simple binary search method in BASIC programs for efficient data handling.

For a demonstration, type in, save, and run "Jump Search" below. Program 1 is a general version for Commodore, IBM, Apple, and the TRS-80 Color Computer. For the Atari, make the line changes listed in Program 2. For the TI-99/4A, one small change is needed to use Program 1. TI BASIC does not allow variables as arguments in DIM statements, so line 110 should be replaced with the following:

```
110 DIM S$(10), PP(10)
```

If you have another computer not mentioned above, use Program 1; it should run with little or no modification.

The demo program creates a list of ten city names in alphabetical order, with population figures for each city (of course, an actual program would contain much more data). Lines 100–140 store the city names in a string array and the population figures in a matching numeric array. (On the Atari, the string array is simulated by manipulating substrings within a single string variable, since there are no true string arrays in Atari BASIC.) Once this is done, you can find the population of any city in the list by searching for its name. For example, if your search finds that AKRON is stored in array element S$(2), then the population for Akron can be found in the numeric array element PP(2).

The city names are stored in the array in alphabetical order because *this search technique works only on data that has been arranged in alphabetical or numeric order.* If you consider the situation for a moment, you'll realize that no organized searching method can speed up the hunt for a particular item in a randomly arranged set of data. If you can't tell whether a word you've found should come before or after the word you're looking for, then you'll have to examine every word in the list until you find an exact match. Arranging the data into alphabetical or numeric order, called *sorting,* is a separate problem and has been considered in previous articles. Just remember that only ordered data can be searched efficiently.

The simplest way to find a word in an alphabetical list is to start at the A's and hunt forward through the alphabet until you find a match. A sequential search of this type is very easy to program (all you need is a FOR-NEXT loop), but it's also slow and inefficient. When the target word is toward the end of the alphabet, sequential searching wastes a lot of time looking through all the preceding words.

Jump To The Center

The binary search method (called *binary* because it repeatedly divides the data list in half) is much faster. Rather than starting at the beginning of the alphabet, it jumps in at the center. Let's look at the example program to see how this works.

The variable B stands for the beginning of the word list, E stands for the end, and C represents the center. Say that your target word is ATLANTA. When the search begins, line 200 finds the center of the ten-word list and jumps to that position (in this case finding the sixth word, ANAHEIM). Since ANAHEIM doesn't match ATLANTA, the program skips to line 250 for a critical test.

At this point the database is divided into two blocks, lower and higher. The program first decides which block holds the target word, then jumps to the center of that block to continue the search. Since ATLANTA comes after ANAHEIM in the alphabet, it must be stored in the higher block of words. Note that in just one step, you've eliminated the need to look at anything in the first half of the database. A sequential search (which compares ATLANTA to ABILENE, then to AKRON, then to ALBANY, etc.) takes six steps to accomplish the same result.

Now it's time for the second jump. Lines 260–270 set a new beginning point just above the center (B = C + 1) and go back to line 200. The program finds the center of the new list (which consists of four words, ANCHORAGE to AUSTIN) and jumps to that position. This time the target word matches the found word. While the binary method found the target word with only two comparisons, a sequential search would require nine (eight comparisons to eliminate ABILENE through ATHENS, and a ninth to confirm ATLANTA).

The more data you have, the more time the binary method saves. For instance, if the list contains 1,000 words, most words are found in about eight comparisons (the sequential method usually requires hundreds). If you expand the list to 10,000 words, only about twelve comparisons are required (compared to thousands for the sequential method). The secret lies in the halving technique. By repeatedly chopping the list in half, this method quickly eliminates large chunks of data from consideration and zeros in on the target. Of course, you're not limited to string data. With slight modifications this routine can search numeric data as well

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: Jump Search (General Version)

```
100 N=10
110 DIM S$(N),PP(N)
120 FOR I=1 TO N
130 READ S$(I),PP(I)
140 NEXT I
150 E=N
160 B=1
170 P=0
180 PRINT "ENTER CITY"
190 INPUT C$
200 C=INT((E+1-B)/2)+B
210 IF E-B<3 THEN 300
220 IF C$<>S$(C) THEN 250
230 P=C
240 GOTO 340
250 IF C$<S$(C) THEN 280
260 B=C+1
270 GOTO 200
280 E=C-1
290 GOTO 200
300 FOR I=B TO E
310 IF C$<>S$(I) THEN 330
320 P=I
330 NEXT I
340 IF P<>0 THEN 370
350 PRINT "DATA NOT FOUND."
360 GOTO 150
370 PRINT S$(P),PP(P)
380 GOTO 150
999 REM CITY & POPULATION DATA
1000 DATA ABILENE,89000
1010 DATA AKRON,237000
1020 DATA ALBANY,250000
1030 DATA ALBUQUERQUE,332000
1040 DATA ALVERINA,29000
1050 DATA ANAHEIM,219000
1060 DATA ANCHORAGE,174500
1070 DATA ATHENS,150000
1080 DATA ATLANTA,425000
1090 DATA AUSTIN,346000
```

Program 2: Atari Line Changes

```
110 DIM C$(15),S$(N*15),P
    P(N):S$=" ":S$(N*15)=
    S$:S$(2)=S$
130 READ C$,A:S$((I-1)*15
    +1,I*15)=C$:PP(I)=A
190 INPUT C$:L=LEN(C$)
220 IF C$<>S$((C-1)*15+1,
    (C-1)*15+L) THEN 250
250 IF C$<S$((C-1)*15+1,(
    C-1)*15+L) THEN 280
310 IF C$<>S$((I-1)*15+1,
    (I-1)*15+L) THEN 330
370 PRINT S$((P-1)*15+1,P
    *15),PP(P)
```

©

128 Sound And Music

Part 2

Philip I. Nelson
Assistant Editor

The second installment of this two-part article explores the Commodore 128's FILTER, SOUND, and PLAY commands and includes three short demonstration programs.

In Part 1 (COMPUTE!, August 1985), we discussed the Commodore 128's VOL, TEMPO, and ENVELOPE commands as well as the basics of sound envelopes and waveforms. This month we'll examine the three remaining sound commands: FILTER, SOUND, and PLAY. Since your 128 *User's Guide* explains the fundamentals, we'll focus on less obvious features and note how these complex commands interact with one another.

FILTER Needs PLAY

Like the ENVELOPE command (see Part 1), FILTER does nothing noticeable until you turn the filter on with a PLAY statement. Insert X1 inside the PLAY string wherever you want to turn the filter on, and X0 where you want to turn it off. If you leave out the X parameter, PLAY ignores preceding FILTER commands (the filter remains off). In the simplest case (a FILTER command followed by PLAY"X1"), the filter affects all three voices. However, you can also filter each voice individually:

```
FILTER 1000,1,0,0,15
PLAY "V1 X1 V2 X0 V3 X0
```

These statements turn the lowpass filter on for voice 1 and turn it off for voices 2 and 3. The 128 remembers which voice to filter when it executes subsequent PLAY statements (more about multivoice music is explained below). However, you can use only one filter *setting* at a time. For instance, you can't use a low-pass filter for voice 1 and a band-pass filter for voice 2. Whenever X1 appears in a PLAY string, the 128 uses the most recent FILTER setting. If no FILTER command has been executed, this may result in silence.

A FILTER Editor

As with other sound effects, the best way to learn is to listen and experiment; Program 1 below, "128 FILTER Editor," lets you do just that. It's self-prompting, so you need only type it in, save a copy, and run it. The menu screen displays all the current filter parameters and lets you change whatever you like. To select any option, press a number key from 0 to 9 and follow the prompts. The program begins with no filtering (all filters off) for comparison.

Option 9 switches you to the display screen, plays an ascending musical scale with whatever filtering you've selected, and displays the FILTER statement currently in effect. Once you find a filter setting you like, write down the FILTER statement displayed on the screen and use it in your own programs. From this screen the number keys 1–6 select different octaves for the scale. Press the space bar to return to the main screen.

Option 7 lets you select any of the 128's ten predefined instrument envelopes, and option 8 controls the tempo at which the scale is played. Note that some of the predefined envelopes don't work well at fast tempos: The note ends before the sound envelope can complete its natural cycle. Use a slower tempo to slow things down and study a particular effect.

The SID filter is a bit notorious. While it works fine on some machines (my old 64 has a great one), its performance may vary from one SID chip to the next. The manual for our preproduction 128 notes that filtering "cannot be counted on," suggesting that nothing was done to improve the 128's filter. With practice you should be able to achieve satisfactory effects on your own machine, though they might sound somewhat different on another computer.

The SOUND Command

SOUND is a very powerful command intended for sound effects rather than music. Unlike PLAY (which defaults to maximum volume), SOUND has a default volume setting of zero. Thus, you must turn the volume up with VOL before the first SOUND statement in a program. And whereas PLAY delays the rest of your program until it completes the current PLAY string, SOUND statements play "in the background" while the program continues. To demonstrate, enter NEW and press RUN/STOP–RESTORE (to clear the SID chip), then type in and run the following two-line program:

```
10 VOL15:SOUND 1,5000,200:SOUN
   D 2,4000,200:SOUND 3,3000,2
   00
20 FORJ=1TO10:PRINT"PROGRAM CO
   NTINUING":NEXT:PRINT"DONE"
```

Notice how the three-voice sound continues even after this program ends and returns the computer to READY mode.

The first number in a SOUND statement (1, 2, or 3) picks one of the 128's three voices. By using different voice numbers, you can play up to three sounds at once. However, the 128 ordinarily waits until a voice has finished the current SOUND statement before starting a new SOUND statement *for that voice*. To illustrate, in line 10 of the above program, change the 2 and 3 to 1; then run it again. Now voice 1 plays three notes in sequence.

In most cases SOUND's background-playing ability is desirable: Sound effects don't slow down the rest of your program. However, in other cases you might want to interrupt a sound immediately (if, for example, the user wants to exit the program). Fortunately, this is easy to do: SOUND statements with zero duration take effect immediately, whether or not preceding sounds have finished. Thus, SOUND 1,0,0 silences voice 1; use FOR J=1 TO 3: SOUND J,0,0: NEXT to silence all three voices.

Since variables can be used for any SOUND parameter, you can create more dynamic, integrated effects by incorporating other program variables in SOUND commands. For example, say that your game uses the variable X to represent a spaceship's screen position. To make a cruising sound, you might substitute something like X*1000 for the frequency number in a SOUND command.

A SOUND Editor

"128 SOUND Editor," listed below, lets you experiment with SOUND commands and design sound effects for your own programs using up to three voices at once. Type in and save Program 2, then run it. The first thing you'll hear are three complex, multivoice sound effects (don't worry if they're not exactly to your taste—you'll soon know enough about SOUND to replace them with your own). Next, the editing screen appears, displaying ten options and all the current SOUND parameters (your *User's Guide* explains the meaning of each parameter). To choose an option, press a number key from 0 to 9. The program instructs you how to proceed and does not let you enter inappropriate values.

Option 1 lets you switch from one voice to another. Option 9 switches you to the display screen, which plays the current sound and displays the SOUND statements that create it. It's fun to experiment with 128 SOUND Editor, and it can save a lot of programming time. Use it to design exactly the sound you want, then copy the SOUND statements from the display screen and use them in your programs. (Though the program can play sounds with one, two, or three voices at once, it's not necessary to use multiple voices. Zero-duration SOUND statements produce no sound and may be ignored.)

The PLAY Command

Designed for real music-making, PLAY is the most versatile of all the 128's sound commands. As outlined in the *User's Guide*, PLAY works much like the familiar PRINT statement. Each PLAY command is followed by a string containing special control characters. The letters A–F are interpreted as notes; thus, the statement PLAY"C D E F" plays the four notes C-D-E-F. In the last example PLAY was followed by a string of characters enclosed in quotation marks. However, PLAY can also handle string variables (A$="C D E F": PLAY A$).

To see this method at work, type in and save Program 3, "128 PLAY Demonstrator." It plays a short, Bach-like tune with several different instrument envelopes. Note that all of the music control characters are stored in DATA statements. Line 50 READs each line of data into a string named A$, and the subroutine at line 20 PRINTs each music string just before it is PLAYed.

Like other strings, PLAY strings can be concatenated (combined) with the + operator, and manipulated with any of the string-related functions: MID$, LEFT$, RIGHT$, LEN, VAL, CHR$, ASC, and STR$. Program 1 contains several different examples.

For complex music you might want to store PLAY strings in a string array. For instance, the following statement stores 100 elements of music data in a string array named M$( ): FOR J=1 TO 100: READ M$(J): NEXT. Once the music array is created, you can quickly access any string it contains: PLAY M$(3) plays the third music string held in M$( ), and so on. This is very helpful for repeating certain passages. You may also find it useful to create separate arrays for different purposes (one to store notes, another for duration characters, and so forth).

Multivoice Music

Since the SID chip has three voices, PLAY can play up to three notes simultaneously. The V control character (followed by 1, 2, or 3) determines which voice is affected. Thus, the statement PLAY "V1 C V2 E V3 G" plays a simple three-note chord. After processing V1 C, the 128 "looks ahead" to see whether it should play other notes at the same time; however, the computer looks ahead only *as far as the next note*. Thus, the statement PLAY "V1 CDE V2 CDE" does not play the notes C-D-E simultaneously with two voices. Instead, it plays two sequential notes (C-D) with voice 1, then two simultaneous notes (E and C) with voices 1 and 2, followed by two sequential notes (D-E) with voice 2.

When all voices play notes of the same duration, multivoice music is not particularly difficult to write: Insert V1 before each note for voice 1, V2 before each voice 2 note, and so forth (concatenations like A$="V1"+A$ can help condense the otherwise cumbersome code). However, when different voices play notes of different durations, you must make sure that all the durations add up.

For instance, you might want voice 1 to hold a long whole note while voice 2 plays a series of sixteenth notes. To keep the timing straight, you should not let voice 1 play another note until voice 2 has finished the equivalent of a whole note (16 sixteenths or whatever). Similarly, the timing may be thrown off if voice 2 plays *more* than 16 sixteenths before voice 1 gets back in the act. The M control character supposedly tells the 128 to wait until all voices finish the current measure before moving ahead. But M is just an adjuster. It can't magically repair music that doesn't add up in the first place.

Interactions

As noted throughout this article, certain 128 sound commands work with certain others. The VOL command, for instance, is needed only for SOUND statements (PLAY sets volume independently with the U control character). TEMPO, FILTER, and ENVELOPE, on the other hand, seem designed to work with PLAY. TEMPO is irrelevant to SOUND (which sets its own duration and so on); ENVELOPE and FILTER have no effect until activated by PLAY.

However, other interactions are possible (at least on our 128, admittedly a preproduction model). For instance, though the SOUND statement provides no way to turn on the filter, SOUNDs can be affected by "leftover" filter settings. If the 128 executes a FILTER statement followed by PLAY"X1", the filter remains on and affects subsequent SOUND statements. PLAY"X0" turns the filter off for SOUND as well as for PLAY.

This interaction can be viewed either as an advantage—filtering is otherwise unavailable with SOUND—or as a pitfall for unwary programmers. To prevent unwanted interactive effects, begin sound and music programs by setting all sound parameters at zero or default values. Commodore 64 programmers often clear the SID chip with FOR J=54272 TO 54296: POKE J,0: NEXT. Though this statement does clear the 128's SID chip, it doesn't necessarily change the 128's sound settings, which are recorded elsewhere in memory.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: 128 FILTER Editor

```
100 GOSUB570:GOTO310
110 FORJ=1TO3:SOUNDJ,0,0:NEXT:
    FILTER0,0,0,0,0:RETURN
120 PLAY A$:RETURN
130 LP$=" OFF":IFLP=1THENLP$="
     {RVS}ON {OFF}"
140 RETURN
150 BP$=" OFF":IFBP=1THENBP$="
     {RVS}ON {OFF}"
160 RETURN
170 HP$=" OFF":IFHP=1THENHP$="
     {RVS}ON {OFF}"
180 RETURN
190 PRINTD$"SET CUTOFF FREQUEN
    CY (0-2047)"
200 INPUTA:IFA<0ORA>2047THENGO
    SUB550:GOTO190
210 FQ=A:RETURN
220 LP=ABS(LP=0):RETURN
230 BP=ABS(BP=0):RETURN
240 HP=ABS(HP=0):RETURN
250 PRINTD$"SET FILTER RESONAN
    CE (0-15)":INPUTA:IFA<0ORA
    >15THENGOSUB550:GOTO250
260 RE=A:RETURN
270 PRINTD$"CHOOSE SOUND ENVEL
    OPE (0-9)":INPUTA:IFA<0ORA
    >9THENGOSUBS50:GOTO270
2B0 WV$="T"+CHR$(A+48):RETURN
290 PRINTD$"CHOOSE TEMPO (1-25
    5)":INPUTA:IFA<1ORA>255THE
    NGOSUB550:GOTO290
300 TM=A:RETURN
310 PRINT"{CLR}{RVS} 128 FILTE
    R EDITOR ":PRINT
320 PRINT"1 {RVS} FREQUENCY
    {OFF}"FQ"{LEFT}{4 SPACES}"
330 PRINT"2 {RVS} LOW
    {2 SPACES}PASS {OFF}";:GOS
    UB130:PRINTLP$
340 PRINT"3 {RVS} BAND PASS
    {OFF}";:GOSUB150:PRINTBP$
350 PRINT"4 {RVS} HIGH PASS
    {OFF}";:GOSUB170:PRINTHP$
360 PRINT"5 {RVS} RESONANCE
    {OFF}";RE"{LEFT} ":PRINT"
    {2 SPACES}{RVS}-----------
    {OFF}"
370 PRINT"7 {RVS} ENVELOPE
    {2 SPACES}{OFF} "MID$(WV$,
    2)T$(VAL(MID$(WV$,2)))
380 PRINT"B {RVS} TEMPO
    {S SPACES}{OFF}"TM"{LEFT}
    {2 SPACES}":PRINT"9 {RVS}
    {SPACE}PLAY{6 SPACES}{OFF}
    ":PRINT"0 {RVS} QUIT
    {6 SPACES}{OFF}{DOWN}"
390 PRINT"{RVS}ENTER YOUR CHOI
    CE (0-9)":PRINT"{3 SPACES}
    {UP}"
400 GETKEYA$:IFA$<"0"ORA$>"9"O
    RA$="6"THENPRINT:GOSUBSS0:
    PRINT:GOTO390
410 IFA$="9"THEN440
420 IFA$="0"THENEND
430 ONVAL(A$)GOSUB190,220,230,
    240,250,250,270,290:PRINTE
    $:GOTO320
440 PRINTCHR$(147)"OCTAVE "MID
    $(OC$,2)CHR$(13)
450 PRINT"LOW{2 SPACES}PASS "L
    P$:PRINT"BAND PASS "BP$:PR
    INT"HIGH PASS "HP$:PRINT
460 PRINT"{RVS}CURRENT FILTER
    {SPACE}STATEMENT:":PRINT:P
    RINT"FILTER ";
470 PRINTMID$(STR$(FQ),2)","MI
    D$(STR$(LP),2)","MID$(STR$
    (BP),2)",";
480 PRINTMID$(STR$(HP),2)","MI
    D$(STR$(RE),2):PRINT:FILTE
    R FQ,LP,BP,HP,RE
490 PRINT"PRESS {RVS} 1 - 6
    {OFF} FOR OCTAVE"CHR$(13)S
    PC(6)"{RVS} SPACE {OFF} TO
     EXIT"
500 F$="X0 ":IFLP=1ORBP=1ORHP=
    1THENF$="X1 "
510 A$=F$+WV$+"S":GOSUB120:TEM
    PO TM
520 GET B$:IFB$=CHR$(32)THENGO
    SUB110:GOTO310
530 IFB$=>"1"ANDB$<="6"THENOC$
    ="O"+CHR$(VAL(B$)+48):PRIN
    T"{HOME}"SPC(6)VAL(B$)
540 A$=OC$+"CDEFGAB":GOSUB120:
    GOTO520
```

```
550 GOSUB110:FORJ=1TO3:SOUNDJ,
    1000+J*500,15,0,0,0,2,J*10
    00:NEXT
560 PRINT"{UP}{RVS}INAPPROPRIA
    TE":SLEEP1:PRINT"{UP}
    {13 SPACES}{3 UP}":RETURN
570 PRINTCHR$(14)CHR$(B):FORJ=
    54272TO54296:POKEJ,0:NEXT:
    VOL15:D$=CHR$(19)
580 FORJ=1TO15:D$=D$+CHR$(17):
    NEXT:FQ=1000:LP=0:BP=0:HP=
    0:RE=15:WV$="T7":TM=55
590 FORJ=1TO35:X$=X$+CHR$(32):
    NEXT:E$=D$+X$+CHR$(13)+X$+
    CHR$(19)+CHR$(13)
600 FORJ=0TO9:READX$:T$(J)="
    {2 SPACES}"+X$:NEXT:OC$="O
    3":GOSUB110:RETURN
610 DATA"PIANO{6 SPACES}","ACC
    ORDION{2 SPACES}","CALLIOP
    E{3 SPACES}","DRUM
    {7 SPACES}","FLUTE
    {6 SPACES}"
620 DATA"GUITAR{5 SPACES}","HA
    RPSICHORD","ORGAN
    {6 SPACES}","TRUMPET
    {4 SPACES}","XYLOPHONE
    {2 SPACES}"
```

Program 2: 128 SOUND Editor

```
10 GOSUB30:GOSUB570:GOTO320
20 PRINT"{CLR}{RVS}128 SOUND E
   DITOR":PRINT:RETURN
30 FORJ=1TO3:SOUNDJ,0,0:NEXT:R
   ETURN
40 PRINTD$"CHOOSE VOICE (1-3)"
   :INPUTA:IFA<1ORA>3THENGOSUB
   550:GOTO40
50 VC=A:RETURN
60 PRINTD$"CHOOSE FREQUENCY (0
   -65535)"
70 INPUTA:IFA<0ORA>65535THENGO
   SUB550:GOTO60
80 FQ(VC)=A:RETURN
90 PRINTD$"CHOOSE DURATION (60
   0=10 SECONDS)"
100 INPUTA:IFA<0THENGOSUB550:G
    OTO90
110 DU(VC)=A:RETURN
120 PRINTD$"CHOOSE DIRECTION O
    F SOUND SWEEP"
130 PRINT"0=UP{2 SPACES}1=DOWN
    {2 SPACES}2=OSCILLATE":INP
    UTA:IFA<0ORA>2THENGOSUB550
    :GOTO120
140 DI(VC)=A:RETURN
150 PRINTD$"CHOOSE MINIMUM FRE
    QUENCY FOR"
160 PRINT"SOUND SWEEP (0-65535
    )":INPUTA:IFA<0ORA>65535TH
    ENGOSUB550:GOTO150
170 IFA=>FQ(VC)THENGOSUB550:GO
    TO150
1B0 MI(VC)=A:RETURN
190 PRINTD$"CHOOSE STEP VALUE
    {SPACE}FOR SOUND SWEEP"
200 PRINT"(LESSER OF 32767 OR"
    FQ(VC)-MI(VC)+1"{LEFT})"
210 INPUTA:IFA<0ORA>32767THENG
    OSUB550:GOTO190
220 IFA>(FQ(VC)-MI(VC))THENGOS
    UB550:GOTO190
230 SV(VC)=A:RETURN
240 PRINTD$"CHOOSE WAVEFORM
    {SHIFT-SPACE}{5 SPACES}0=T
    RIANGLE"
250 PRINT"1=SAWTOOTH{2 SPACES}
    2=PULSE{2 SPACES}3=WHITE N
    OISE"
260 INPUTA:IFA<0ORA>3THENGOSUB
    550:GOTO240
270 WV(VC)=A:RETURN
280 PRINTD$"CHOOSE PULSE WIDTH
    "
290 PRINT"(0-4095)":INPUTA:IFA
    <0ORA>4095THENGOSUB550:GOT
    O280
300 PW(VC)=A:RETURN
310 GOSUB20
320 PRINT"1 {RVS} VOICE
    {6 SPACES}{OFF}"VC:PRINT"2
     {RVS} FREQUENCY{2 SPACES}
    {OFF}"FQ(VC)"{LEFT}
    {4 SPACES}"
330 PRINT"3 {RVS} DURATION
    {3 SPACES}{OFF}"DU(VC)"
    {LEFT}{4 SPACES}"
340 PRINT"4 {RVS} DIRECTION
    {2 SPACES}{OFF}"DI(VC)DI$(
    DI(VC))
350 PRINT"5 {RVS} MINIMUM
    {4 SPACES}{OFF}"MI(VC)"
    {LEFT}{4 SPACES}":PRINT"6
    {SPACE}{RVS} STEP VALUE
    {OFF}"SV(VC)"{LEFT}
    {4 SPACES}"
360 PRINT"7 {RVS} WAVEFORM
    {3 SPACES}{OFF}"WV(VC)WV$(
    WV(VC))
370 PRINT"8 {RVS} PULSEWIDTH
    {OFF}"PW(VC)"{LEFT}
    {4 SPACES}"
380 PRINT"9 {RVS} HEAR SOUND
    {OFF}":PRINT"0 {RVS} QUIT
    {7 SPACES}{OFF}":PRINT
390 PRINT"{RVS}ENTER YOUR CHOI
    CE (0-9)":PRINT"{3 SPACES}
    {UP}"
400 GETKEYA$:IFA$<"0"ORA$>"9"T
    HENPRINT:GOSUB550:PRINT:GO
    TO390
410 IFA$="9"THEN440
420 IFA$="0"THENGOSUB30:END
430 ONVAL(A$)GOSUB40,60,90,120
    ,150,190,240,280:PRINTE$:G
    OTO320
440 PRINT"{CLR}THE FOLLOWING S
    OUND STATEMENTS":PRINT"
    {2 SPACES}CREATE THE SOUND
    S YOU HEAR."
450 PRINT"ZERO-DURATION SOUNDS
     ARE SILENT."
460 FORJ=1TO3:SOUNDJ,FQ(J),DU(
    J),DI(J),MI(J),SV(J),WV(J)
    ,PW(J):NEXT
470 FORJ=1TO3:PRINT:PRINT"SOUN
    D ";
480 PRINTMID$(STR$(J),2)","MID
    $(STR$(FQ(J)),2)","MID$(ST
    R$(DU(J)),2)",";
490 PRINTMID$(STR$(DI(J)),2)",
    "MID$(STR$(MI(J)),2)","MID
    $(STR$(SV(J)),2)",";
500 PRINTMID$(STR$(WV(J)),2)",
    "MID$(STR$(PW(J)),2):NEXT
510 PRINT:PRINT"PRESS {RVS}RET
    URN{OFF} TO EXIT":PRINTSPC
    (6)"{RVS}SPACE {OFF} TO RE
    PEAT"
520 GETKEYA$:IFA$=CHR$(13)THEN
    GOSUB30:GOTO310
530 IFA$=CHR$(32)THENGOSUB30:G
    OTO440
540 GOTO520
550 GOSUB30:FORJ=1TO3:SOUNDJ,1
    000+J*500,15,0,0,0,2,J*100
    0:NEXT
560 PRINT"{UP}{RVS}INAPPROPRIA
    TE":SLEEP1:PRINT"{UP}
    {13 SPACES}{3 UP}":RETURN
570 PRINTCHR$(14):D$=CHR$(19):
    FORJ=54272TO54296:POKEJ,0:
    NEXT:FORJ=1TO15
5B0 D$=D$+CHR$(17):NEXT:GOSUB2
    0:VOL15:FORJ=1TO38:X$=X$+C
    HR$(32):NEXT
590 VC=1:E$=D$+X$+CHR$(13)+X$+
    CHR$(13)+X$+CHR$(19)+CHR$(
    13)
600 FORK=2000TO4000STEP220:FOR
    J=1TO3:SOUNDJ,K*2+J*20,45,
    2,K,K/3,2,4095-K
610 NEXTJ,K:FORJ=45TO1STEP-5:S
    OUND1,J*1000,5,1,J*100,J*2
    80,2,2300
620 SOUND2,3200-J*20,5,0,0,0,2
    ,1500:SOUND3,J*1200,5,1,J*
    120,J*300,2,3000
630 NEXT:FORJ=1TO3:SOUNDJ,1000
    0,200,1,J*2000,J*400,2,230
    0:NEXT:FORJ=1TO3
640 READFQ(J),DU(J),DI(J),MI(J
    ),SV(J),WV(J),PW(J):NEXT:F
    ORJ=0TO3:READA$
650 WV$(J)="--- "+A$:NEXT:FORJ
    =0TO2:READA$:DI$(J)="--- "
    +A$:NEXT:RETURN
660 DATA10000,260,2,2000,60,2,
    2000,0,0,0,0,0,0,2000,0,0,
    0,0,0,0,2000
670 DATA"TRIANGLE","SAWTOOTH",
    "PULSE{3 SPACES}","NOISE
    {3 SPACES}"
680 DATA"UPWARD{3 SPACES}","DO
    WNWARD ","OSCILLATE"
```

Program 3: 128 PLAY Demonstrator

```
10 GOTO30
20 PRINTA$:PLAYA$:RETURN
30 PRINTCHR$(147)CHR$(14)SPC(3
   )CHR$(18)"128 PLAY DEMONSTR
   ATOR"CHR$(13)
40 FORJ=54272TO54296:POKEJ,0:N
   EXT:FILTER0,0,0,0:FORJ=1TO3
   :SOUNDJ,0,0:NEXT
50 READA$:IFA$<>"Z"THENGOSUB20
   :GOTO50
60 PRINT:PRINTSPC(2)CHR$(1B)"P
   RESS P TO PLAY AGAIN, Q TO
   {SPACE}QUIT"
70 GETKEYG$:IFG$="P"THENRUN
80 IFG$<>"Q"THEN70
90 END
100 DATA U15 X0 V1 S
110 DATA T7 O5 C O4 B O5 IC SO
    4 GRERGR
120 DATA T6 CDC O3 B O4 IC SO3
     GRERGR
130 DATA T7 CGDGEGDGC
140 DATA O4 C O3 BAGFEDC
150 DATA O5 C O4 BAGFED
160 DATA T6 CGDGEGFGEGDG
170 DATA CG O3 #A O4 G O3 A O4
     G O3 G O4 G
180 DATA O3 F R O5 FE I F S DR
     O4 BR O5 DR
190 DATA T2 G O6 G O5 A O6 G O
    5 B O6 G C O6 GDGFG
200 DATA ERDCDGC O5 B
210 DATA T4 ERDCDGC O4 B
220 DATA T6 ERDCDGC O3 B
230 DATA T0 ERDCDGC O2 BC
240 DATA T7 O3 CDEFGABC
250 DATA O4 CDEFGABC
260 DATA O5 CDEFGAB
270 DATA O6 CR O5 CR I O3 CR
50000 DATA Z
```

©

EASY Apple Screen Editing

Roland Brown

Here's a way to make BASIC programming easier and more fun: an advanced screen editor that makes up for the Apple's lack of full-screen editing. COMPUTE! *published an earlier version of this utility, "BASIC Line Editor," in February 1983. This month's all-new version has been updated and enhanced to work on any Apple II-series computer (including the Apple IIc) with DOS 3.3 or ProDOS, in 80-column as well as 40-column mode.*

Although Applesoft BASIC is a powerful language, its screen editor leaves much to be desired. Some Apple II owners invest in a ROM editor, others write their programs with a word processor, and the rest just suffer with the frustrating ESCape codes. But ROM editors cost money, word processors don't let you flip back and forth between the text editor and BASIC to test changes, and suffering isn't *always* good for the soul. So here's a better solution: "BASIC Line Editor," a powerful utility that lets you easily modify BASIC program lines.

To prepare the BASIC Line Editor, type in and save the program listed below. It's a BASIC filemaker that POKEs the machine language program into memory, then BSAVEs it to disk as a binary file (named BLE2 to distinguish it from BLE, the original version of the program).

Once you've run the filemaker, you're ready to use the BASIC Line Editor. Start it by typing BRUN BLE2 and pressing RETURN. The program loads at memory address $2000, then checks to see which operating system is present before moving itself to a safe location. (Note that this process can destroy part of a long BASIC program. If you have a long BASIC program in memory, you should save it *before* you activate the BASIC Line Editor.)

Now you're ready to put the Editor to work. To edit a BASIC program line, type & followed by the desired line number. For instance, enter &100 to edit line 100. The BASIC Line Editor displays the line on the screen in a format somewhat different than Applesoft's. The line is continuous rather than centered on the screen, there are no extra spaces in the line except between quotation marks, and all control characters are displayed in inverse video.

Editing Commands

The BASIC Line Editor provides 13 new editing functions. Most are accessed by pressing the CTRL (Control) key together with a letter key. Here's a quick reference table followed by a detailed description of each command:

CTRL-B	**block back**
CTRL-C	**convert hex to decimal**
CTRL-D	**delete right**
CTRL-F	**block forward**
CTRL-H	**cursor left**
CTRL-I	**insert**
CTRL-M	**return**
CTRL-S	**search**
CTRL-T	**truncate**
CTRL-U	**cursor right**
CTRL-V	**verbatim**
DELETE	**delete left**
ESC	**return to BASIC**

CTRL-B (block back) moves the cursor back to the previous colon, or if there is no previous colon, to the beginning of the line.

CTRL-C (convert hex) converts hexadecimal numbers to decimal. This command moves the cursor above the line being edited, prints a $ prompt on the screen and waits for you to enter a number. This value is converted to decimal and printed. Then the cursor returns to its original position on the line.

CTRL-D (delete right) deletes the character under the cursor. The cursor stays where it is and everything to the right moves back one space.

CTRL-F (block forward) moves the cursor forward to the next colon, or if there is no colon, to the end of the line.

CTRL-H (cursor left) moves the cursor back one space.

CTRL-I (insert) puts the BASIC Line Editor in insert mode. Any characters you type are inserted in the line until you use another Editor command.

CTRL-M (return) is the same as pressing RETURN. No matter where the cursor is located on the line, pressing CTRL-M enters the line into the program.

CTRL-S (search) searches for the next character entered.

CTRL-T (truncate) truncates the line at the cursor position (deletes everything after the cursor). The cursor ends up one space beyond the new end of the line.

CTRL-U (cursor right) moves the cursor forward one space.

CTRL-V (verbatim) lets you enter control characters verbatim. If the keypress immediately after CTRL-V is a CTRL key combination, it is interpreted as a control character rather than as a BASIC Line Editor command. CTRL-V is useful for adding RETURN (CTRL-M) or backspace (CTRL-H) characters to a line for improved printing control. If the keypress immediately following CTRL-V is not a CTRL key combination, CTRL-V has no effect. Remember that the BASIC Line Editor shows control characters in reverse video.

DELETE (delete left) deletes the character to the left of the cursor and moves the cursor back one space. (The DELETE key is found only on the IIe and IIc.)

ESC (return to BASIC) puts you back in BASIC. If you make a mistake when editing a line with the BASIC Line Editor, press ESC to exit back to BASIC without losing the line.

Program Notes

Activating the Editor resets the stack to the same level as does BASIC, sets up the ampersand vector ($3F5), moves the DOS buffers downward to protect DOS, and restarts BASIC. The Editor uses existing BASIC routines to read the input line and find the desired line in memory. If you try to edit a line that doesn't exist, the Editor simply returns to BASIC. If the line is found, its contents are read and listed on the screen. Text characters are listed just as they are stored. When the Editor finds a token (an encoded BASIC keyword), it locates the word in the BASIC keyword table and lists it on the screen.

Once the Editor lists the line, it enters editing mode. This part of the program gets a command from the keyboard, processes it, and updates the screen. Space doesn't permit a detailed explanation of how each Editor command works. If you're familiar with Apple machine language programming, you may find it interesting to trace through the various routines on your own.

BASIC Line Editor

Version By Tim Victor, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
53 8Ø FOR I = 8192 TD 9157: READ
       A: PDKE I,A: NEXT
8Ø 9Ø PRINT CHR$ (4);"8SAVE 8LE2
      ,A$2ØØØ,L$3C6": END
1E 1ØØ DATA 173,Ø,191,2Ø1,76,2Ø8
       ,13,169,3,32
68 11Ø DATA 245,19Ø,24,165,116,1
       Ø5,4,76,27,32
A8 12Ø DATA 56,165,116,233,3,133
       ,116,133,2Ø7,141
2D 13Ø DATA 175,32,165,115,133,2
       Ø6,141,174,32,169
2Ø 14Ø DATA 177,133,235,169,32,1
       33,236,16Ø,Ø,177
A8 15Ø DATA 235,145,2Ø6,23Ø,2Ø6,
       2Ø8,2,23Ø,2Ø7,23Ø
85 16Ø DATA 235,2Ø8,2,23Ø,236,16
       5,235,2Ø1,7Ø,2Ø8
75 17Ø DATA 234,165,236,2Ø1,35,2
       Ø8,228,177,235,23Ø
1F 18Ø DATA 235,2Ø8,2,23Ø,236,14
       1,176,32,17,235
19 19Ø DATA 24Ø,41,173,176,32,24
       ,1Ø9,174,32,133
8E 2ØØ DATA 2Ø6,177,235,23Ø,235,
       2Ø8,2,23Ø,236,1Ø9
ØD 21Ø DATA 175,32,133,2Ø7,24,17
       7,2Ø6,1Ø9,174,32
B3 22Ø DATA 145,2Ø6,2ØØ,177,2Ø6,
       1Ø9,175,32,145,2Ø6
43 23Ø DATA 136,24Ø,2ØØ,173,174,
       32,141,246,3,173
F8 24Ø DATA 175,32,141,247,3,169
       ,76,141,245,3
FA 25Ø DATA 16Ø,11,185,162,32,32
       ,24Ø,253,136,16
5E 26Ø DATA 247,96,141,217,196,1
       93,197,21Ø,16Ø,178
1E 27Ø DATA 197,2Ø4,194,141,56,3
       2,32,32,12,218
6E 28Ø DATA 32,26,214,176,1,96,1
       Ø4,1Ø4,32,156
5D 29Ø DATA 252,16Ø,2,177,155,2Ø
       Ø,17Ø,177,155,32
2C 3ØØ DATA 36,237,16Ø,6,14Ø,123
       ,5,132,2Ø6,165
99 31Ø DATA 37,141,151,2,165,155
       ,133,235,165,156
CB 32Ø DATA 133,236,16Ø,4,177,23
       5,2ØØ,2Ø1,Ø,24Ø
44 33Ø DATA 44,16,36,162,2Ø8,142
       ,68,Ø,142,69
4E 34Ø DATA Ø,41,127,17Ø,173,255
       ,255,48,17,224
96 35Ø DATA Ø,2Ø8,3,32,72,1,238,
       68,Ø,2Ø8
48 36Ø DATA 239,238,69,Ø,2Ø8,234
       ,2Ø2,16,243,32
5Ø 37Ø DATA 72,1,56,176,2Ø5,16Ø,
       6,169,192,141
62 38Ø DATA 152,2,132,2Ø7,32,34,
       1,32,12,253
1E 39Ø DATA 2Ø1,255,2Ø8,2,169,12
       8,2Ø1,16Ø,144,81
5E 4ØØ DATA 44,152,2,48,15,112,6
       5,141,7Ø,1
8C 41Ø DATA 32,35,2,169,192,141,
       152,2,48,218
ØC 42Ø DATA 112,34,72,164,2Ø7,13
       2,227,164,2Ø6,14Ø
5F 43Ø DATA 149,2,2ØØ,32,236,1,1
       32,2Ø7,32,96
86 44Ø DATA 1,2Ø6,149,2,198,2Ø7,
       164,227,196,2Ø7
E5 45Ø DATA 2Ø8,242,32,34,1,1Ø4,
       32,11Ø,1,164
75 46Ø DATA 2Ø7,196,2Ø6,2ØØ,144,
       3,32,236,1,76
F7 47Ø DATA 1Ø5,Ø,164,2Ø7,169,19
       2,141,152,2,48
8C 48Ø DATA 157,44,152,2,48,13,8
       Ø,24Ø,162,192
2A 49Ø DATA 142,152,2,73,192,2Ø1
       ,64,2Ø8,213,162
DD 5ØØ DATA 192,142,152,2,2Ø1,14
       1,24Ø,12,2Ø1,155
77 51Ø DATA 24Ø,46,164,2Ø7,32,25
       3,1,76,1Ø5,Ø
88 52Ø DATA 16Ø,Ø,132,2Ø7,32,34,
       1,32,155,1
1D 53Ø DATA 73,128,16,2,41,63,16
       4,2Ø7,153,Ø
87 54Ø DATA 2,2ØØ,196,2Ø6,2Ø8,23
       2,169,Ø,153,Ø
4E 55Ø DATA 2,16Ø,1,162,255,76,6
       8,212,164,2Ø6
7Ø 56Ø DATA 32,34,1,16Ø,Ø,24Ø,23
       5,72,173,151
11 57Ø DATA 2,133,37,152,197,33,
       144,6,229,33
61 58Ø DATA 23Ø,37,176,246,133,3
       6,141,123,5,32
33 59Ø DATA 34,252,1Ø4,96,132,2Ø
       7,32,34,1,32
32 6ØØ DATA 155,1,2Ø1,7Ø,96,14Ø,
       15Ø,2,9,128
17 61Ø DATA 2Ø1,16Ø,176,2,73,192
       ,32,11Ø,1,164
EØ 62Ø DATA 2Ø6,2ØØ,32,236,1,172
       ,15Ø,2,96,172
Ø9 63Ø DATA 149,2,32,34,1,32,155
       ,1,164,2Ø7
A8 64Ø DATA 32,34,1,141,153,2,16
       5,37,72,173
5F 65Ø DATA 123,5,133,36,72,173,
       153,2,32,24Ø
6D 66Ø DATA 253,1Ø4,2Ø5,123,5,2Ø
       8,7,197,36,165
F5 67Ø DATA 36,141,123,5,1Ø4,144
       ,7,197,37,2Ø8
4B 68Ø DATA 3,2Ø6,151,2,173,153,
       2,96,173,123
DF 69Ø DATA 5,172,179,251,192,6,
       2Ø8,22,44,31
87 7ØØ DATA 192,16,17,141,1,192,
       72,56,1Ø1,32
8D 71Ø DATA 74,144,3,44,85,192,1
       Ø4,1Ø5,Ø,74
18 72Ø DATA 168,177,4Ø,44,84,192
       ,96,192,Ø,24Ø
FF 73Ø DATA 37,32,247,1,132,2Ø7,
       132,227,32,15
34 74Ø DATA 2,14Ø,149,2,196,2Ø6,
       24Ø,13,32,96
3E 75Ø DATA 1,238,149,2,23Ø,2Ø7,
       172,149,2,2Ø8
D1 76Ø DATA 239,164,2Ø7,32,236,1
       ,164,227,96,132
A9 77Ø DATA 2Ø6,32,34,1,32,156,2
       52,164,2Ø6,96
Ø7 78Ø DATA 192,Ø,24Ø,1,136,96,1
       62,11,2Ø2,48
Ø6 79Ø DATA 25Ø,221,127,2,2Ø8,24
       8,189,138,2,141
66 8ØØ DATA 14,2,176,255,196,2Ø6
       ,24Ø,1,2ØØ,96
22 81Ø DATA 169,128,44,169,Ø,44,
       169,64,141,152
36 82Ø DATA 2,96,169,186,141,7Ø,
       1,164,2Ø7,196
ØE 83Ø DATA 2Ø6,24Ø,6,2ØØ,32,61,
       1,2Ø8,244,164
D5 84Ø DATA 2Ø7,96,169,186,141,7
       Ø,1,164,2Ø7,24Ø
52 85Ø DATA 6,136,32,61,1,2Ø8,24
       6,164,2Ø7,96
16 86Ø DATA 172,151,2,136,132,37
       ,32,34,252,169
9E 87Ø DATA Ø,141,123,5,32,156,2
       52,162,Ø,169
56 88Ø DATA 164,32,11Ø,1,32,12,2
       53,157,Ø,2
23 89Ø DATA 232,2Ø1,141,2Ø8,242,
       32,199,255,32,167
62 9ØØ DATA 255,169,189,32,24Ø,2
       53,165,63,166,62
72 91Ø DATA 32,36,237,164,2Ø7,96
       ,128,132,136,149
43 92Ø DATA 148,137,147,15Ø,134,
       13Ø,131,179,188,232
69 93Ø DATA Ø,221,6,9,12,18,38,5
       6,35,Ø
89 94Ø DATA 59,Ø,62,Ø,77,Ø,8Ø,Ø,
       85,Ø
B9 95Ø DATA 93,Ø,1Ø3,Ø,1Ø8,Ø,124
       ,Ø,131,Ø
81 96Ø DATA 134,Ø,139,Ø,153,Ø,15
       7,Ø,162,Ø
E4 97Ø DATA 165,Ø,176,Ø,18Ø,Ø,19
       Ø,Ø,193,Ø
DF 98Ø DATA 2ØØ,Ø,2Ø5,Ø,214,Ø,22
       5,Ø,238,Ø
6Ø 99Ø DATA 241,Ø,248,Ø,251,Ø,28
       ,1,36,1
3C 1ØØØ DATA 64,1,67,1,73,1,84,1
        ,9Ø,1
F1 1Ø1Ø DATA 93,1,97,1,1ØØ,1,1Ø3
        ,1,1Ø8,1
F4 1Ø2Ø DATA 111,1,123,1,149,1,1
        52,1,199,1
73 1Ø3Ø DATA 2Ø6,1,2Ø9,1,216,1,2
        19,1,224,1
AE 1Ø4Ø DATA 231,1,239,1,3,2,8,2
        ,11,2
2F 1Ø5Ø DATA 3Ø,2,36,2,46,2,56,2
        ,64,2
F7 1Ø6Ø DATA 72,2,93,2,Ø,Ø          ©
```

WHITE HOUSE COMPUTER

P.O. Box 4025
Williamsport, PA 17701

1-800-351-3442

1-717-322-7700

HOURS: Monday thru Friday 9:00 am till 6:00 pm

VISA and MC Accepted 4%

PRINTERS

PANASONIC

1090	187.00
1091	**237.00**
1092	389.00
1093	585.00
3151	455.00

C.ITOH

8510A	285.00
8510BC1	329.00
8510BC2	389.00
8510BP1	339.00
8510SP	389.00
8510SR	429.00
1550BCD	485.00
1550P	445.00
A10-20P	465.00
F10-40P/S	875.00
F10-55P/S	1059.00
F10 Tractor	169.00
F10 Sheet Feeder	349.00

STAR MICRONICS

SG10	**214.00**
SG15	379.00
SD10	345.00
SD15	450.00
SR10	485.00
SR15	585.00
Powertype	309.00

EPSON

RX80	219.00
LX80	**245.00**
FX80	369.00
JX80	529.00
RX100	359.00
FX100	490.00

LEGEND

880	209.00
1080	229.00
1380	269.00
1385	305.00

C-64

MPS 801	179.95
802	208.95
803	175.95
MCS 801	314.95
DPS 1101	314.95

ATARI

XTM 201 Non-Impact Dot Matrix	99.95
XTC 201 Color Non-Impact Matrix	109.95
XDM 121 Daisy Wheel	209.95
XMM 801 Dot Matrix Impact	169.95
STC 504 Color/ Non-Impact	139.95
STD 121 Daisy Wheel	219.95
SMM 801 Dot Matrix/ Impact	279.95

CITIZEN

MSP-10	325.00
MSP-15	479.00
MSP-20	469.00
MSP-25	629.00

OKIDATA

182	219.95
84	640.95
92	**349.95**
93	560.95

CARDCO

LO1	365.00
LQ3	**209.00**

INTERFACES

MPP 1150	52.00
850 (Atari)	108.00
UPrint w/16K Buffer	79.95
Cardco G	55.95
Cardco B	37.95

PRINTER PAPER

Lazor 500 shts	11.95
Lazor 1000 shts	15.95

COMPUTERS

COMMODORE

C-64	149.95
C-128	275.95

ATARI

800 XL	99.95
130 XE	145.95
130 ST	379.95
520 ST	CALL

SOFTWARE

BRODERBUND "ATARI" & COMMODORE

Print Shop	29.95
Graphics Library	18.95
BS Writer	41.95
Stealth	21.95

VIDEO CASSETTE RECORDER

$295.95

Symphonic VHS/Remote

- ●105 Channel Cable Ready
- ●14 Day/1 Program Timer
- ●12 Pre-set Channels
- ●Auto Rewind

DISK DRIVES

ATARI

Indus GT	**215.00**
1050	165.95
"Happy" 1050	365.95
MSD Dual	459.00

C-64

Indus GT	239.00

MODEMS

C-1600	44.95
C-1650	54.95
C-1660	54.95
C-1670 300/1200 Band	185.95
C-64 MPP-1064	59.95
MPP 1000E	69.95
Telelearning	39.95
Hayes 300	189.95
Hayes 1200	385.95
Mitey Mo	59.95
Westridge	69.95
Compuserve Starter Kit	21.95

DISKETTES

SKC

SS/SD	10.50
SS/DD	13.50
DS/DD	15.50

MAXWELL

MD-1	15.95
MD-2	21.95

ELEPHANT

SS/SD	13.50
SS/DD	15.50
DS/DD	19.50

MONITORS

ZENITH

122A	82.95
123G	75.95

TEKNIKA

MJ-10 Composite & Separate Video	199.95
MJ-22 RGG Composite & Separate Video	279.00

AMDEK

300G	119.00
300A	129.00
Color 300	235.00
Color 500	345.00

XTRON

Comcolor I	189.00

COMMODORE

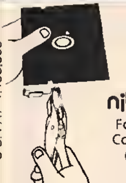
C-1702	199.95
C-1802	204.95
C-1902 RGB	259.95
C-1901 Monochrome	129.95

ATARI

XC 141 14" Composite Color	199.95
SM 124 12" Monochrome/ High Res	135.95
SC 1224 12" RGB Color	335.95

POLICY: No deposit on COD orders. FREE freight on all prepaid cash orders over $300 in the continental USA. APO and FPO add $5 per hundred. For priority mail add $8 per hundred. PA residents add 6% salas tax. Delective products must have Prior RA number. Schools net 15.

IT'S LIKE FREE DISKETTES



U.S. PAT 4,488,358

Your 5¼" single side disks are usable on the other side. You paid for one side, why not use the other... **IT'S FREE!**

Nibble Notch will **open** your **new** disk. It's easy... won't harm existing data. Try it!

nibble notch I

For Apple, Franklin, Commodore & Atari (w/Atari Drives); square notch.

only $14.95* PLUS P&H

nibble notch II

For all other computers; square notch & index hole.

only $21.90* PLUS P&H

ALSO

DISK OPTIMIZER II©

Apple II Series Software

Pro DOS • DOS 3.3 • Pascal

Examines your **new** disk, locks out bad sectors and **certifies it 100% ERROR-FREE** in 30 seconds or less! Also checks drive speed... and more!

SUPER SAVER PACKAGE

Nibble Notch I and Disk Optimizer Combo (Optimizer alone reg. $24.95)

$29.95* FOR BOTH!

QUALITY DISKETTES low as 99¢

*add $2 ($5 frgn) for P & H. Fl. Res. add 5% Sales Tax

Toll Free 1-800-642-2536

FL 1-305-748-3770
OR SEND CHECK OR
MONEY ORDER TO:

VISA

MasterCard

nibble notch® computer products

4211 NW 75th TERRACE,• DEPT. 662 LAUDERHILL, FL 33319

PROFESSIONAL HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

GLD. Thoroughbred "Gold" Edition™
A "Full" tealured Ihoroughbred analysis designed for Ihe piofessional and Ihe serious novice
$159.95 complete

EGLD. Enhanced "Gold" Edition™
"Gold " Edition with complete Masler Bettor™ syslem integrated onto the same disk. This powerlul program will transfer all horses and scores to Ihe bel analysis with a "single keystroke " (Master Bettoi ™ included) **$199.95** complete

GLTD. Limited "Gold"™
Enables **Prolessional Handicappers** lo assign specific values to the iacing vaiiables "Ihey" feel aie important. Cieale program weight based on a particulai Irack and fine tune il foi **maximum** win peicenlage This piogiam is designed foi "ease of use" The usei needs **no programming experience.**
(contains Integialed Bettoi™) **$299.95** complete

GO. Gold Oog Analysis™
The "ONLY" professional gieyhound analysis available Ihal evaluates ALL vaiiables
$149.95 w/integrated Bettoi **$199.95**
Limited Veision **$299.95**

MHH. Master Harness Handicapper™
Piofessional software designed to piovide a thoiough analysis of all Irotter and pacer iaces in North Ameiica and Canada.
$159.95 complete
$199.95 w/Integiated Bettor Limited **$299.95**

Protessor Plx Football™
Complete STATISTICAL ANALYSIS on Dala Base allowing "Designated" previous games lo be evalualed
Statistical Series **$99.95**
w/Win-Loss Powei Ratings **$149.95**

NBA. Basketball™
NBA **$99.95** w/college **$129.95**
w/powei ratings **$149.95**

LOT. Lottery Analysis™
Statistical compaiison piogiam designed lo delect subtle patterns in winning numbeis and digits.
Lottery (3-4 digit) **$79.95**
w/Lotto (Max. 99 Digit) **$99.95**

PC-3 Portable Computer (4k) with choice of Thoioughbred, Greyhound or Tiottei™ **$249.95** (Includes portable computei and piogiam.)

M-100 Portable (24k) w/choice of Thoroughbred, Greyhound, or Trotter. **$649.95** (Includes portable computei and piogiam)

BROCHURE AVAILABLE

IBM™
APPLE™
TRS-80™
CPM™
COMMOOORE™

Prof. Jones
1940 W. State St.
Boise, ID 83702

VISA MasterCard

48 HR. FREE SHIPPING

☎ CALL
208-342-6939

TERMS· FREE SHIPPING ALL SOFTWARE. Add $6.00 hardware / $6 00 C O D. / UPS Blue $6 00 / Out of Counlry $9 00 / ID Residenls 4% / 3 weeks peisonal checks / Cash pnce only. add 2% Visa, MC / Pnces subject to change

Lyco Computer Marketing & Consultants

SAVE ON THESE IN STOCK PRINTERS

AXIOM

GP550AT (Atari)	239
GP550CD (C-64	239
GP550PC (IBM)	229
GP550AP (Apple)	269
GP700AT (Atari)	439
GP700AP (Apple)	439
ELITE5CD (C-64)	229
GP100AT (Atari)	CALL

C. ITOH

PROWRITER 8510AP	279
8510AP +	299
8510BPI	309
8510SP	375
8510SR	425
8510SCP	449
8510SCR	469
7500AP	205
7500AR	245
1550P	439
1550AP +	459
1550BCP	469
A10-30P	445
F10-40P or S	859
F10-55P or S	1039
F10 Tractor	159
F10 Sheet Feeder	249

CARDCO

32K BUFFER (C-64)	59

CITIZEN

MSP-10	289
MSP-15	449
MSP-20	449
MSP-25	569
Sheetfdr (10/20)	189
Sheetfdr (15/25)	199

CORONA

LP300 Laser Printer	2699
200361 Toner Cartridge	89

EPSON

LX80 (NEW)	225
FX80 +	349
FX100 +	479
JX80 Color	489
RX100	369
HOMEWRITER 10 (New)	CALL
CR220 (C-64)	159
LQ1500 Par	999
LQ1500 Ser	1059
HI80 Color Plotter	379

JUKI

Juki 6100	375
RS232 Serial Board	55
8100 Trector	119
6100 Sheet Feeder	209
Juki 6300	765

OKIDATA

Okimate 10	CALL
Okimate 20	CALL
182	215
192	349
193	565
92P	349
93P	565
84P	645
92 Imagewriter	349

*IBM versions also

BROTHER

HR-15XL-P	359
HR-15XL-S	359
HR-35P	839
HR-35S	839
2024L-P	949
M1009-P	189

LEGEND

880	199
1080	225
1380	265
1385	299

DIGITAL DEVICES

16K BUFFER	75
32K BUFFER	89
64K BUFFER	125

TOSHIBA

P1340	559
P351	1289

MICROBITS

Microstuffer	99

DIABLO

D25	549
630API	1599
630ECS	1759
D801F	2395
P32CQ1	699
P38	1749
C150	999

PANASONIC

1091	***239***
1090	185
1092	375
1093	429
3151 Letter	429
4K Buffer	65

SILVER REED

EXP400	249
EXP500	295
EXP550	409
EXP770	749

STAR MICRONICS

SG-10	***215***
SG-15	375
SD-10	339
SD-15	445
SR-10	485
SR-15	585
Powertype	305
SB-10	CALL
SG-10C-64 (NEW)	CALL

OVER 2000 SOFTWARE TITLES IN STOCK!

★ PRINTER ★ INTERFACING

AXIOM

AT846 (Atari)	65
Parallax (C-64)	65

MICROBITS

MPP-1150 (Atari)	54
MPP-1150XL (Atari)	59
MicroPrint (Atari)	39
64K RAM Board (600xl)	65

DIGITAL DEVICES

ApeFace XLP (Atari)	49
ApeFace 12XLP (1200xl)	49
U-Print A (Atari)	54
U-Print C (C-64)	49
U-Print A16K Buffer	74
U-Print A64K Buffer	99
U-Print AP16K (Apple)	99
U-Print AP64K (Apple)	139
U-CALL RS232 (Atari)	37

MICROTEK

Dumpling GX (Apple)	59
Dumpling 16K (Apple)	89
RV-611C (Apple)	49

TYMAC

Connection (C-64)	55
Tackler (Apple)	49
PPC-100 (Apple)	39

ORANGE MICRO

GRAPPLER + (Apple)	85
Grappler 16K (Apple)	149
ORANGE (Apple)	59
Grappler CD (C-64)	79

CARDCO

G-Wiz (C-64)	54
C/? + G (C-64)	45
C/?PS (C-64)	49
C/?B (C-64)	39
C/?AT (Atari)	49
C/?AP (Apple)	49
C/?IBM (IBM)	29

PRINTING PAPER

3000 SHEETS FANFOLD	$42 75
1000 SHEETS FANFOLD	$19 75
1000 SHEET LETTER	$21 95
200 SHEETS LETTER	$8 99
150 RAG STATIONARY	$10 99
MAILING LABELS (1 in)	$9 95
14 x 11 1000 FANFOLD	$24 75

INNOVATIVE CONCEPTS

Flip-n-File 10	3 50
Flip-n-File 15	8 25
Flip-n-File 25 Lock	17.95
Flip-n-File 50	17 25
Flip-n-File 50 Lock	22 95
Flip-n-File Rom	17 25

WICO Joysticks

15-9714 Bal Handle	16.75
50-2030 Boss	11.99
50-2002 Super 3-Way	19.99
50-0110 IBM-Analog	29 95
50-1030 Apple-Analog	29 95
72-4545 Trackball	29 95

DRIVES

MSD

SD1 Drive (C-64)	229
SD2 Drive (C-64)	469

INDUS

GT ATARI	***215***
GT COMMODORE	235

DISKETTES

SKC (Box 10)

(Qty)	10	100	500
5¼ SSDD	13 75	12 99	11 99
5¼" DSDD	15 75	14 99	14 25

ELEPHANT (Box 10)

5¼" SSSD	13 50	12 99	11 99
5¼" SSDD	14 99	13 99	12 99
5¼" DSDD	16 99	15 99	14 99
Premium SSDD	15 95	14 95	13 95
Premium DSDD	17 95	16 95	15 95

MAXELL (Box 10)

5¼" MD-1	16 75	15 75	14 75
5¼" MD-2	22 99	21 99	20 99

VERBATIM (DataLife) (Box 10)

5¼" SSDD	19 99	18 99	17 99
5¼" DSDD	24 99	23 99	22 99

NO LABEL DISKETTES

NL 5¼" SSDD	10 99 (Box 10)
NL 5¼" DSDD	15 99 (Box 10)

- Free Diskette Writer Pen!
- Free Storage Case!
- Lifetime Warranty!

IBM-PC SOFT-WARE

*LOTUS

Lotus 1-2-3	309 00
Symphony	439 00

*ASHTON-TATE

Framework	385 00
d Base II	289 00
d Base III	385 00
d Base II upgrade	135 00

BRODERBUND (IBM)

The Print Shop	CALL
Bank St Writer	49 95
Ancient Art of War	29 95
Loderunner	21 95

IBM-PC COMPATABLE

PARADISE

Multi-Display Card	289
Modular Graphics Card	275
Five-Oak	155

TECHMAR

First Mate-64K	249
Captain-64K	199
Second Mate	209
Captain JR-128K	309
Graphics Master	465

OMEGA

10 Meg Bernoulli Box (IBM)	1899
5 Meg Bernoulli Box (MAC)	1499

TALLGRASS TECHNOLOGIES

25mb drive	CALL
35mb drive	CALL
50mb drive	CALL
80mb drive	CALL

AST

Six Pack Plus 64K	CALL
Mega Plus II 64K	269
Memory 128K (mega)	199
Memory 256K (Mega)	335
Mono-Graph Plus	339
Advantage-AT-128K	389

*QUADRAM

Quad Jr Exp Chassis	519 00
Quad Jr Exp Memory	209 00
Quad Memory Jr	209 00

*LEADING EDGE

Nutshell	69 95
LEWP Basic	65 00
LEWP Merge Print	99 00
LE Spell Correction	169 00

Zenith

Z-15D	Call

Leading Edge

PC Compatable	Call

TOLL FREE 1-800-233-8760

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P O Box 5088
Jersey Shore PA 17740

RISK FREE POLICY

In-stock items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. Volume discounts available. PA residents add sales tax. APO, FPO, and international orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for MasterCard or Visa. Personal checks require 4 weeks' clearance before shipping. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Free catalog with order. All items subject to change without notice.

ATARI

ATARI

800XL	CALL
130XE (NEW)	CALL
520ST (NEW)	CALL
1050 Drive	165
1010 Recorder	42
1020 Printer	55
1025 Printer	159
1027 Printer	179
850 Interface	1'9

ATARI SOFTWARE (NEW)

Codewriter	35.75
Filewriter	20.75
Reportwriter	20.75
Menuwriter	20.75
Home Integrator	19 75
Small Bus. Inventory	11 75
Salesman Expenses	11.75
Accs Rec/Pay	11.75
Retail Invoice	11.75
Final Legecy	15.75
Adventure Writer	18 75
Star Voyeger	11.75

SSI (Atari)

Colonial Conquest	24 75
Field of Fire	24 75
Kampfgruppe	36 75
Objective Kursk	24 75
Market Garden	29 75
Six Gun Shootout	24.75
War in Russia	49 75
Computer Baseball	24 75
Computer Quarterback	24 75
Gemstone Warrior	22 75
50 Mission Crush	24 75
Questron	29 75

SPINNAKER

Delta Drawing Room	19 95
Cosmic Life Room	19 95
Up for Grabs Room	19 95

PERSONAL PERIPHERALS

Super Sketch C-64	29 95
Printer Utility C-64	14 99
Super Sketch Atari	25 95
Super Sketch Apple/Ibm	55 00

SUB LOGIC (Atari)

Flight Simulator II	***32.75***
Night Mission Pinball	20 75

ATARI

850 Interface	109

CONTINENTAL

Home Accountant	44 75
*985 Book of Atari Software	16 95

TRONIX

S A M · Atari	38 50
S A M C-64	38 50

ADVENTURE INT'L (Atari)

Diskey	32 75
Ultradissembler	32.75
Maxi Golf	18.75
Ralley Speedway	26.75
Whomper Stomper	18.75
Adventures 1-12 each	14 50
Adv. Hint Book	5.50

MICROPROSE (Aterl)

Kennedy Approech	21 75
Crusade in Europe	24.75
Decision in Desert	24 75
Solo Flight	20 75
Neto Commender	20 75
Spitfire Ace	18.75
F-15 Strike Eagle	20.75
Hellcat Ace	18 75

SYNAPSE (Atari)

SynFile	33 75
SynCalc (AT/C-64)	33 75
SynComm.	26 75
SynTrend	26 75
SynChron	26 75
SynStock	26 75

BRODERBUND (Atarl)

The Print Shop	28 95
Graphics Librery	18 95
Grephice Librery II	19.50
Bank St. Writer	42.75
Whistler's Brother	18 95
Spelunker	18.95
Stealth	18 95
Serpent's Star	24 95
Mask of the Sun	24.95

SCARBOROUGH (Atari)

NET WORTH (AT/C-64)	48 75
Mastertype	22 75
Improved Mastertype	23 75
Mastertype Filer	22 75

apple

SCARBOROUGH (Apple)

NET WORTH	48.75
Improved Mastertype	22.75
Mastertype Filer	22.75

BROOERBUND (Apple)

The Print Shop	31 50
Graphics Library	18.50
Graphics Library II.	18 95
Paper Refill	11 95
Bank St Writer 64K	41 50
Bank St Writer 128K.	42 75
Bank St Filer	42 75
Bank St Mailer	42.75
Bank St Speller	42.75
Carmen Sandiego	22.75
Karateka	22.75
Captain Goodnight	22.75
Muppey Cruise	25 75

SUB LOGIC (Apple)

Flight Simulator II	***32.75***
Night Mission Pinball	20.75

MICROPROSE (Apple)

Crusade in Europe.	25 75
Decision in Desert	25 75
F-15 Strike Eagle	21 75
Nato Commander	21 75

SSI (Apple)

Mech Brigade	36.75
Norway 1985	22 75
Market Garden	29 75
Six Gun Shootout	24 75
50 Mission Crush	24 75
Phantasie	24 75

Spinnaker

Alphabet	$18 75
Story Machine	$19 75
Kids on Keys	$18 75
Grandma	$19 75
Snooper Troop	$22 75

COMMODORE

COMMODORE

C-128 NEW	CALL
1571 Drive	CALL
1572 Drive	CALL
1902 Monitor	CALL
1670 Modem	CALL
C-64 Computer	CALL
1541 Drive	189
MPS801 Printer	LOW
1702 Monitor	199
Simon's Basic	24 75
Assembler 64	34.75
Super Expander	22.75
Logo 64	49.75
Pilot 64.	38.75
Easy Cale	34.75
Easy Script	38 75

SSI (C-64)

Colonial Conquest	24.75
Wings of War	24 75
Computer Ambush	34 75
Field of Fire	24.75
Fighter Command	36.75
Kampfgruppe	36.75
Mech Brigade	36 75
Market Garden	29 75
Six Gun Shootout	24 75
Computer Baseball	24 75
Computer Quarterback	24.75
Imperium Galactum	24.75
Phantasie	24 75
Cartels & Cutthroats	24.75
50 Mission Crush	24.75
Questron	24.75

SCARBOROUGH (C-64)

Build A Book	24 75
Improved Mastertype	23 75
NET WORTH	48 75
Mastertype Filer	22 75

SUB LOGIC (C-64)

Flight Simulator II	***32.75***
Night Mission Pinball	20.75

CARDCO

Digitizer Camera	189.95
32K Printer Buffer	59.95
Numeric Keypad	34.95
CB/5 5-slot Board(64).	54.00
CB/2 2-slot Boerd(64).	25.00
S'More Basic Rom	49 95
Write Now-64	35.00
Mail Now-64	29 00
Spell Now-64	29 00
File Now-64	29.00
Paint Now-64	29.00
Calc Now-64	29 00
Tax Survival	29 00
Super Printer Utility	27 95
Write Now-Vic-20	29 95

HES

HES Games 84	22 95
Omni Writer/Spell	34 95
HES Mon 64	23 95
Microsoft Multiplan	55 00
Type N Write	19 95
Turtle Graphics II	23 95
Cell Defense	22 95
Paint Brush	12 95
Tri Math	22 95
Graphics Basic	27 95
HES Kit	29 95
Millionaire	23 95
64 Forth	24 95
HES Writer 64	24 95

BATTERIES INCLUDED

Paper Clip	59 95
Spell Pak	34 95
Consultant	59 95
Paper Clip w/Spell Pak	75 95
Home Pak	34 95
Bus Card	129 95
80 Column Board	109 95

BRODERBUND (C-64)

The Print Shop	***28.75***
Graphics Library	18 75
Graphics Library II	19 75
Karateka .	19 75
Castles Dr Creep	19 75
Bank St Writer	32 75
Loderunner	20 75
Mask of the Sun	24 75
Spelunker	19 75
Serpent's Star	24 75
Whistler's Brother	18 75
Raid Bungeling Bay	18 75

MICROPROSE (C-64)

Kennedy Approach	21 75
Crusade in Europe	24 75
Decision in Desert	24 75
Solo Flight	20 75
Nato Commander	20 75
Spitfire Ace	18 75
F-15 Strike Eagle	20 75
Hellcat Ace	18 75

MODEMS

US ROBOTICS

Password 1200M	229
Password 1200F	229
Password 300M	139
Password 300F	139
Autodial 212A	359
PCM5	319
PCM64	519
PCM256	769
S-100	379
Courier	469
Microlink	469
Telpac Ms-DOS	79

RACAL-VADIC

2400PC	549
2400PA	799
2400V	559
1200PC	329
300V	205
300PC	199
George Software	69

HAYES

SMARTMODEM 300	139
SMARTMODEM 1200	379
SMARTMODEM 1200B	349
SMARTMODEM 2400	599
MICROMODEM IIE	135

TELE LEARNING

Total Telecommunications (C-64)	29 95
AP-250 (300 Baud Apple)	69 95
IB-250 (300 Baud IBM)	69 95

ANCHOR

Volksmodem	55
Volksmodem 12.	186
Mark 12	229

NOVATION

IBM-300/1200 MS-DOS ext	319
IBM-300/1200 CPM-86 ext	319
IBM 300/1200/2400 ext	529
IBM 300/1200/2400 MS-OOS	579
IBM MS-DOS int	325
IBM CPM-86 int	325
Macmodem 300/1200	315
Macmodem 300/1200/2400	585
Apple Cat II	219
Upgrade Apple Cat II	225
Cat 300 Acoustic	139
J-Cat RS232	89
WESTRIOGE MODEM (C-64)	59
MITEY MO MOOEM (C-64)	69
Compuserve	19 85

★ ***SUPER SPECIAL!*** ★ ★ ***LOWEST PRICES!*** ★

MONITORS

AMDEK

300 Green	125
300 Amber	139
310 Amber IBM	155
Color 300 Audio	245
Color 500 Composite	369
Color 600	429
Color 700	495
Color 710	569

NEC

JB-1260 Green	95
JB-1201 Green	135
JC 1215 Color	235
JC 1216 RGB	375
JC1460 Color	265
JB-1205 Amber	139

PRINCETON GRAPHICS

MAX-12 Amber	189
HX-12 RGB	475
SR-12 RGB	599

SAKATA

SG1000 12" Green	99
SA1000 12" Amber	109
SG1500 12" Green TTL	119
SA1500 12" Amber TTL	129
SC100 13" Color Comp	209
SC200 13" RGB	389
STSI Tilt Stand	29

PANASONIC

DT 1300 RG1 composite	299

ZENITH

ZVM 122A Amber	84
ZVM 123G Green	75
ZVM 124 Amber IBM	129
ZVM 131 Color	275
ZVM 133 RGB	389
ZVM 135 Composite	449
ZVM 136 Hi Res Color .	589

X-TRON

Comcolor I Composite Green	185

TAXAN

115	12	Green Composite	CALL
116	12	Amber Composite	CALL
121	12"	Green TTL	135
122	12"	Amber TTL	145
220	14	Color Composite	259
410	12	RGB Hi Res IBM	329
420	12	RGB Super Hi IBM	409
440	12	RGB Ultra Hi Res	569
Tilt Stand			35

AMERICA'S MAIL ORDER HEADQUARTERS

LYCO COMPUTER

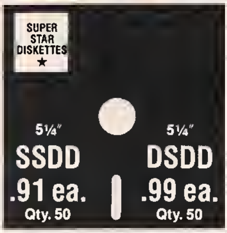
WORLD'S LEADER IN SALES & SERVICE

TO ORDER
CALL TOLL FREE
800-233-8760
In PA 1 717-327-1824
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

What the world really needs is a 99 cent Double Sided, Double Density Diskette with a LIFETIME WARRANTY!

And DISK WORLD! has it.

Introducing Super Star Diskettes: the high quality diskette with the lowest price and the best LIFETIME WARRANTY!



Super Star diskettes are sold in multiples of 50 only. Diskettes are shipped with white Tyvec sleeves, reinforced hubs, user ID labels and write-protect tabs.

In the course of selling more than a million diskettes every month, we've learned something: higher prices don't necessarily mean higher quality.

In tact, we've found that a good diskette manufacturer simply manufactures a good diskette...no matter what they charge for it. (By way ot example, consider that none of the brands that we carry has a return rate of greater than 1/1,000th of 1 percent!)

In other words, when people buy a more expensive diskette, they aren't necessarily buying higher quality.

The extra money might be going toward flashier advertising, snazzier packaging or simply higher profits.

But the extra money in a higher price isn't buying better quality.

All of the good manufacturers put out a good diskette. Period.

How to cut diskette prices ...without cutting quality.

Now this discovery posed a dilemma: how to cut the price of diskettes without lowering the quality.

There are about 85 companies claiming to be "diskette" manufacturers.

Trouble is, most of them aren't manufacturers.

Rather they are fabricators or marketers, taking other company's components, possibly doing one or more steps of the processing themselves and pasting their labels on the finished product.

The new Eastman Kodak diskettes, for example, are one of these. So are IBM 5¼″ diskettes. Same for DYSAN, Polaroid and many, many other familiar diskette brand names. Each of these diskettes is manutactured in whole or in part by another company!

So, we decided to act just like the big guys. That's how we would cut diskette prices...without lowering the quality.

We would go out and find smaller companies to manufacture a diskette to our specifications...specitications which are higher than most...and simply create our own "name brand" diskette.

Name brand diskettes that otfered high quality at low prices.

Boy, did we get lucky. Our Super Star Diskettes are the same ones you've been using for years...without knowing it.

In our search for the low priced, high quality diskette of our dreams, we found something even more interesting.

We found that there are several manutacturers who don't give a hoot about the consumer market for their diskettes. They don't spend millions of dollars in advertising trying to get you, the computer user, to use their diskettes.

Instead, they concentrate their efforts on turning out the highest quality diskettes they can...because they sell them to the software publishers, computer manufacturers and other folks who (in turn) put their name on them...and sell them for much higher prices to you!

After all, when a software publisher or computer manufacturer or diskette marketer puts their name on a diskette, they want it to work time after time, everytime. (Especially sottware publishers who have the nasty habit ot copy-protecting their originals!)

Super Star Diskettes. You already know how good they are. Now you can buy them...cheap.

Well, that's the story.

Super Star diskettes don't roll off the boat from Pago-Pago or emerge from a basement plant just east ot Nowhere.

Super Star diskettes have been around for years...and you've used them for years as copy-protected software originals, unprotected originals. Sometimes, depending on which computer you own, the system master may have been on a Super Star diskette. And maybe more than once, you've bought a box or two or more of Super Star diskettes without knowing it. They just had some "big" company's name on them.

Super Star Diskettes are good. So good that a lot of major sottware publishers, computer manufacturers and other diskette marketers buy them in the tens or hundreds ot thousands.

We buy them in the millions.

And than we sell them to you.

Cheap.

When every little bit counts, it's Super Star Diskettes.

You've used them a hundred times...under different names.

Now, you can buy the real McCoy, the same diskette that major software publishers, computer manufacturers and diskette marketers buy...and call their own.

We simply charge less.

Super Special!

Order 50 Super Star Diskettes and we'll be happy to sell you an Amaray Media-Mate 50 for only $8.75, shipping included...a lot less than the suggested retail price of $15.95.

Regular OISK WORLD! price: $9.69 ea. + $2.00 Shpng.

DISKETTE STORAGE CASES

DISK CADDIES

The original flip-up holder for 10 5¼″ diskettes. Beige or Grey only.

$1.65 ea. + .20 Shpng.

DISKETTE 70 STORAGE

Dust-free storage for 70 5¼″ diskettes. Six dividers included. An excellent value.

$11.95 ea. +$3.00 Shpng.

HOW TO ORDER:

ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)
INQUIRIES:
1-312-256-7140

FOR FASTEST SERVICE, USE NO-COST MCI MAIL: Our address is DISKWORLD. It's a FREE MCI MAIL letter. No charge to you. (Situation permitting, we'll ship these orders in 24 hours or less.)

SHIPPING: 5¼″ & 3½″ DISKETTES—Add $3.00 per each 100 or fewer diskettes. **OTHER ITEMS:** Add shipping charges as shown in addition to other shipping charges. **PAYMENT:** VISA, MASTERCARD and Prepaid orders accepted. **COD DRDERS:** Add additional $3.00 special handling charge. **APO, FPO, AK, HI & PR ORDERS:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance. We ship only to United States addresses, except for those listed above. **TAXES:** Illinois residents, add 8% sales tax.

MINIMUM ORDER: $35.00 or 20 diskettes.

The Super Star LIFETIME WARRANTY!

Super Star Diskettes are unconditionally warranted against defects in original material and workmanship so long as owned by the original purchaser. Returns are simple: just send the defective diskettes with proof of purchase, postage-paid by you with a short explanation of the problem, and we'll send you the replacements. (Incidentally, cotfee stained diskettes and diskettes with staples driven through them don't qualify as "detective".)

HOURS:
Human: 8AM-6PM Central Time, Monday through Friday
Answering Machine: 6PM-8AM, All Times
MCI MAIL: 24 hours a day.

WE WILL MEET OR BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES SUBJECT TO THE SAME TERMS AND CONDITIONS.

DISK WORLD!, Inc.

629 Green Bay Road
Wilmette, Illinois 60091

Incredible value!

Nashua™ Diskettes

LIFETIME WARRANTY!

97¢ ea. 5¼" SSDD Qty. 50

$1.07 ea. 5¼" DSDD Qty. 50

These are poly-bagged diskettes packaged with Tyvek sleeves, reinforced hubs, user identification labels and write-profect tabs. NASHUA Corporation is a half-billion dollar corporation and a recognized feader in magnefic media.

SOFT SECTOR ONLY! Sold in mulfiples of 50 only!

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Cenfral Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! Authorized Distributor NASHUA MAGNETIC MEDIA

ATHANA DISKETTES

The great unknown!

97¢ ea. ← 5¼" SSDD Qty 50

$1.01 ea 5¼" DSDD → Qty. 50

You've used these diskettes hundreds of times...as copy-protected originals on some of the most popular software packages. They're packed in poly-bags of 25 with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs

LIFETIME WARRANTY!

SOFT SECTOR ONLY! Sold in multiples of 50 only.

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Cenfral Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! Authorized Distributor ATHANA MAGNETIC MEDIA

DISKETTE STORAGE CASES

AMARAY MEDIA-MATE 50: A REVOLUTION IN DISKETTE STORAGE

Every once in a while, someone takes the simple and makes it eleganf! This unit holds 50 5¼" diskettes, has grooves for easy stacking, inside nipples to keep diskettes from slipping and several other feafures. We like it! **$9.69** ea + $2.00 Shpng

DISKETTE 70 STORAGE: STILL A GREAT BUY.

Dusf-free storage for 70 5¼" diskettes. Six dividers included An excellenf value **$11.68** + $3.00 Shpng

DISK CADDIES

The original flip-up holder for 10 5¼" diskettes Beige or grey only **$1.65** ea + 20¢ Shpng

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INOUIRIES:
1-312-256-7140

HDURS: 8AM-5PM Cenfral Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE DN THE SAME PRDDUCTS ANO OUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! The value leader in Computer supplies And sccessories.

FANTASTIC LOW PRICES ON

BASF QUALIMETRIC DISKETTES!

LIFETIME WARRANTY!

$1.14 ea. 5¼" SSDO Qty. 20

$1.31 ea 5¼" OSOO Qty. 20

5¼" SSDD-96TPI → **$1.46** ea.   5¼" DSDO-96TPI → **$1.75** ea.

PACKED IN CARDBOARO CASES!

BASF DUALIMETRIC DISKETTES have a LIFETIME WARRANTY wifh Tyvek sleeves, reinforced hubs, user identification labels and write-profect tabs.

SDFT SECTOR ONLY! MINIMUM DRDER: 2D DISKETTES

BASF 3.5" MICRD-FLDPPIES
5SSDD-135 TPI → **$2.34** ea

BASF 5¼" HIGH DENSITY FDR IBM PC-AT
DSDD-HD → **$3.14** ea

FOR OROERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATIQN & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD! Authorized Reseller Information Processing Media BASF

DISK WORLD!

Ordering & Shipping Instructions

Shipping: 5¼" & 3.5" OISKETTES—Add $3.00 per each 100 or lewer diskettes. **Ofher Items:** Add shipping charges as shown in addition to other shipping charges. **Payment**: VISA and MASTERCARO accepted **COD Orders:** Add addifional $3.00 Special Handling charge **APD, FPO, AK, HI & PR Orders:** fnclude shipping charges as shown **and** additional 5% of total order amount fo cover PAL and insurance. **Taxes:** Illinois residents only, add 7% sales tax

Prices subjecf fo change wifhouf nofice.
This ad supercedes all other ads.
Not responsible for fypographical errors.
MINIMUM TQTAL QRDER: $35.00

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois. 1-312-256-7140)

INFORMATION & INOUIRIES:
1-312-256-7140

HDURS: 8AM-5PM Cenfraf Time
Monday-Friday

WE WILL BEAT ANY NATIDNALLY ADVERTISED PRICE ON THE SAME PROOUCTS AND OUANTITIESf

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD!

PRINTER RIBBONS:

at extraordinary prices!

Brand new nbbons, manufactured fo Original Equipment Manufacturer's specifications, in housings. (Not re-inked or spools only.)

LIFETIME WARRANTYf

Epson MX-70/80	$3.58 ea. + 25¢ Shpng.
Epson MX-100	$4.95 ea. + 25¢ Shpng.
Okidata Micro83	$1.48 ea. + 25¢ Shpng.
Okidata Micro84	$3.66 ea. + 25¢ Shpng.

FOR ORCERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

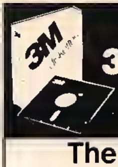
INFORMATION & INOUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Cenfral Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIESf

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD!



Now, the lowest prices ever on

3M Scotch® DISKETTES

LIFETIME WARRANTY!

The best deal on 3M diskettes you've ever seen!

FREE!

FLIP N' FILE 15 with every order for 5.25" SSDD and DSDD.

$1.42 ea. — 5.25" SSDD with FREE Flip n' File 15
5.25" DSDD with FREE Flip n' File 15 — **$1.74** ea.

100% LIFETIME WARRANTY

This is a Super Special promotion. It was supposed to end around the end of May.

But we decided to buy more than 1,000,000 3M diskettes packed in the FREE Flip n' File 15's...and give you the benefits of this terrific value.

One word of warning: **this offer is limited only to supplies on hand. Once these supplies are used up, the prices stay the same...but there's no free Flip n' File.**

The last time we ran an offer like this, everything was sold out in about six weeks.

So don't wait. Order now.

Other 3M diskettes:

(Flip n' File offer does not apply.)

5.25" SSDD-96TPI	$2.06 ea.
5.25" DSDD-96TPI	$2.57 ea.
5.25" DSDD-HD for IBM PC/AT	$3.93 ea.
3.50" SSDD-135TPI for Apple Mac	$2.86 ea.

DATA CARTRIDGES

100% certified 3M data cartridges.

DC-1000	$13.90 ea.
DC-300XLP	$19.83 ea.
DC-600A	$22.13 ea.

Sold in cases of 10 only.
Add $5.00 shipping per 10.

DISK WORLD! Authorized Distributor Information Processing Products 3M presenting . . . **CAPTURE**™

A NEW WAY TO UNLOCK THE POWER OF YOUR C64 OR C128*

- **CAPTURE** IS A CARTRIDGE THAT PLUGS INTO YOUR COMPUTER'S EXPANSION PORT.
- **CAPTURE** DOES NOTHING — UNTIL YOU PRESS ITS **CAPTURE** BUTTON. UNTIL THEN, A RUNNING PROGRAM CANNOT DETECT ITS PRESENCE.
- **CAPTURE** THEN TAKES CONTROL — NO IFS, ANDS OR BUTS — AND PRESENTS A MENU.
- **CAPTURE** WILL NEATLY SAVE EVERYTHING IN YOUR COMPUTER TO YOUR 1541 OR COMPATIBLE DISK DRIVE — ALL 64K OF RAM, CPU, VIC AND SID CHIP REGISTERS — EVERYTHING. IN EASY TO VIEW CHUNKS.
- **CAPTURE** WILL, IF YOU WANT, PRE-CONFIGURE YOUR COMPUTER'S RAM SO THAT ONLY MEMORY ALTERED BY YOUR PROGRAM NEED BE SAVED.
- **CAPTURE** WILL WRITE A BOOT ON YOUR DISK SO YOU CAN RELOAD AND BEGIN EXECUTION RIGHT WHERE YOU LEFT OFF.
- **CAPTURE** DOES ALL THIS AT A COST OF JUST **$39.95**

BUT HERE'S THE BEST PART

- **CAPTURE** WILL MAKE AN AUTO-START CARTRIDGE OF YOUR PROGRAM. IT'S EASY! JUST FOLLOW THE DIRECTIONS ON THE SCREEN. NOW PLUG IN YOUR CARTRIDGE AND TURN ON YOUR COMPUTER. IN LESS THAN TWO SECONDS YOUR PROGRAM BEGINS AGAIN AT PRECISELY THE POINT WHERE YOU **CAPTURE**'D IT. MAGIC!

  BESIDES **CAPTURE,** YOU NEED A *promenade* C1 AND A SUPPLY OF CPR3 CARTRIDGE KITS.

ORDERING INFORMATION

• **CAPTURE** CARTRIDGE — COMPLETE WITH INSTRUCTIONS	$ 39.95
• *promenade* C1 — EPROM PROGRAMMER WITH DISK SOFTWARE	99.50
• CPR3 CARTRIDGE KIT — PC BOARD, CASE AND 3 EPROMS	29.95
• DR—EPROM ERASER, TWO AT A TIME, 3 TO 10 MINUTES	34.95
• STARTER SET — **CAPTURE,** *promenade* C1 AND 1 CPR3	149.95
• DELUXE SET — **CAPTURE,** *promenade* C1, DR AND 2 CPR's	199.95

SHIPPING AND HANDLING — USA: UPS SURFACE $3.00
BLUE LABEL $5.00 NEXT DAY AIR $13.00
CANADA: AIR MAIL $7.00 OTHER FOREIGN AIR $12.00

CALIFORNIA RESIDENTS ADD APPLICABLE SALES TAX
COD ORDERS, USA ONLY, ADD $3.00

C64 AND C128 TM COMMODORE ELECTRONICS, LTD. *WHEN OPERATING IN 64 MODE

TO ORDER: TOLL FREE	800-421-7731	TECHNICAL SUPPORT AND	408-287-0259
FROM CALIFORNIA	800-421-7748	FROM OUTSIDE THE US:	408-287-0264

JASON-RANHEIM
580 PARROT STREET
SAN JOSE, CA USA 95112

VISA MasterCard AMERICAN EXPRESS

ATARI

Atari Hardware Section

ATARI is a trademark of ATARI INC

Atari 130 XE ... Call for Latest
Atari 1050 Disk Drive ... Price

SUPERPRINTER PACKAGES

SG-10 Printer and U-Print A	289
Panasdnic 1091 and U-Print A	318
Legend 880 & U-Print A	287
Citdh 7500AP + and U-Print A	399
Pdwertype & U-Print A	379

Super Printer Packages have no extra charges when shipped in Continental USA.

ATARI INTERFACES

Aid Interfast I	99.95
U-Print A	59.95
U-Print A/16K Buffer	79.95
U-Print A/32K Buffer	99.95
U-Print A/64K Buffer	109.95
R-Verter Adaptor	39.95

INDUS GT

for Atari	219

MPP 1000E

Modem	89.95

NEW
U-Call Modem Adapter 44.95

ATARI SUPER SPECIALS

Balleries Included

80 Cdlumn Carlridge	**49.95**
B-Graph-D	**44.95**
Beachhead-D	**21.95**
Jupiter Mission-D	**31.95**
S.A.M.-D	**37.95**
Home Pak-D	**31.95**
Crusade In Europe-D	**24.95**

Super Special Prices ENeclive Now Through September 31, 1985

ATARI SOFTWARE

BRODERBUND

Bank Street Writer-D	49.95
Loderunner-D	23.95
Spelunker-D	20.95
Stelth-D	20.95
Print Shop-D	34.95
Serpent's Star-D	27.95
Print Shop Paper	16.95
Print Shop Graphics-D	19 95

ELECTRONIC ARTS

Archon	19 95
Archon II	24.95
Mule	19.95
Realm/Impossibility	19.95
Murder/Zinderneut	19 95
Music Construction	19.95
Pinball Construction	19.95
Dne on Dne	24.95
Seven Cities of Gold	24.95
Financial Cookbook	29.95

INFOCOM

Enchanter-D	24.95
Zork I-D	24 95
Hitchiker-D	24.95
Cutthroats-D	24.95
Wishbringer-D	29.95

MICRDPRDSE

F-15 Strike Eagle-D	23.95
Crusade in Europe-D	27.95
Decision in Desert-D	27.95
Kennedy Approach-D	23.95

MISCELLANEDUS ATARI

Flight Simulator II-D	34.95
Megatont II + -D	19 95
Monkey Wrench II-Cart	23 95
Beachhead-D	21 95
Net Worlh-D	49 95
Omnitrend Universe-D	69 95
MMG Compiler-D	69.95
Summer Games-D	27.95
Ultima I-D	23.95
Ultima II-D	37.95
Ultima III-D	37.95
Ultima IV-D	41.95
Letter Perfect-D	39.95
Data Perfect-D	39.95
Micro-Leag. Baseball-D	29.95
Paper Clip-D	39.95

MISC. ATARI (cont'd.)

Dn-Track Racing-D	20.95
Bounty Bob-Cart	29 95
Ramrod XL	79 95
Alien Voice Box	99.95

DSS

Basic XL-Cart	54.95
MAC 65XL-Cart	69.95
Action-Cart	69.95
Writer's Tool-Cart	69.95
MAC 65 Tool Kit-D	27.95
Action Tool Kit-D	27.95
Basic XL Tool Kit-D	27 95

SSI

Carrier Force-D	37.95
Combat Leader-D	24.95
Cosmic Balance-D	24 95
Broadsides-D	24.95
50 Mission Crush-D	24 95
Rails West-D	24.95
Computer Ambush-D	37 95
Computer Baseball-D	24.95
Breakthrough/ Ardennes-D	37.95
Imperium Galatium-D	24 95
Oper. Markel Garden-D	32 95
Kampfgruppe-D	37.95
Computer Duarterback-D	24 95
Colonial Conquest-D	24 95
Gemstone Warrior-D	21 95
Six Gun Shootout-D	24 95

SYNAPSE

Synflie-D	34.95
Syncaic-D	34 95
Syntrend-D	34 95
Synchron-D	27.95
Syncomm-D	27.95
Synstock-D	27 95
Relax-D	79.95
Blue Max 2001-D	23.95

HUNDREDS OF ITEMS AVAILABLE. PLEASE CALL

GENERAL HARDWARE

star micronics inc
THE POWER BEHIND THE PRINTED WORD.

SG-10	**219**
SG-15	**369**
SD-10	**339**
SD-15	**449**
SR-10	**Call**
SR-15	**Call**

PRINTER BUFFERS

Microfazer	Call
U-Print-16K	79.95
U-Print-32K	99.95
U-Print-64K	109.95

MODEMS

Volksmodem 300	59
Volksmodem 1200	189
Hayes 1200	Call
Hayes 300	169
Micromodem IIE	Call
Maxwell 300/1200	319

PRINTERS

Citoh 8510AP +	329
Legend 1080	239
Citoh 7500AP	219
Epson	Call
Toshiba 1351	Call
Toshiba 1340	Call
Legend BB0	219
Panasonic 1090	199
Panasonic 1091	249
Powertype	309

With The Purchase Of Any Printer

Buy

The Print Shop-Apple	**31.95**
The Print Shop- Atari & C-64	**27.95**

Super Special Prices ENective Now Thru September 31, 1985

APPLE

BRDDERBUND

Print Shop-D	33 95
Karateka-D	21.95
Champ Loderunner-D	21.95
Bank Slreet Writer-D	44.95
Print Shop Graphics-D	17 95
Dazzle Draw	39.95

SSI

Gemstone Warrior	21.95
Kamptgruppe	39.95
Phanlasie	25.95
Ouestron	33.95

INFDCOM

Hitchiker	29 95
Zork I, II or III	29 95
Enchanter	29 95
Deadline	34.95
Wisbringer-D	29.95

ELECTRONIC ARTS

Skyfox	29.95
Dne on Dne	29 95
Seven Cities/Gold	29.95
Archon	24.95
Adv Construction Kit	37 95

APPLE MISCELLANEDUS

Ghostbusters	25.95
Flight Simulator II	34.95
F-15 Strike Eagle	21.95
Summer Games	25.95
Micro League Baseball	29.95
Star League Baseball	23.95
Wizardry/Proving	33.95
Ultima III	39.95
Bruce Lee	25.95
Millionaire	33.95
Death/Carribean	21 95
Felony	21.95
Conan	25.95
Sargon III	33.95
Dollars & Sense	64.95
PFS File	79.95
PFS Write	79.95
PFS Report	79.95
Multiplan	84 95
Supercalc III	124.95
Sideways	39 95
Crosstalk	129.95
Jane w/Mouse	129.95
BPI Software	Call

EST. 1982

ComputAbility

P.D. Box 17BB2, Milwaukee, WI 53217
DRDER LINES DPEN
Mon-Fri 11 a.m. - 7 p.m. CST • Sat. 12 p m. - 5 p.m. CST

To Drder Call Toll Free

800-558-0003

For Technical tnfo, Drder Inquiries, or tor Wisc. Drders

414-351-2007

DRDERING INFDRMATIDN: Please specity system. For tast delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 2 weeks lo clear. Charges tor CDD are $3 00 School Purchase Orders welcome In CONTINENTAL USA. include $3 00 shipping per software order Include 4% shipping on all Hardware orders, minimum $4.00 Mastercard & Visa please include card # and expiralion date WI residents please add 5% sales tax. HI, AK, FPD, APD, Canadian orders — add 5% shipping, minimum $5.00 All olher toreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice

COMMODORE 64

SUPER PRINTER PACKAGES

SG-10 & G-WIZ	282
Panasdnic 1091 & G-WIZ	313
Panasdnic 1090 & G-WIZ	261
Legend 880 & G-WIZ	271
Legend 1080 & G-WIZ	291
Citdh 8510AP & G-WIZ	395

Super Printer Packages have no extra charges added when shipped in Continental USA

MODEMS

[illegible]	[illegible]
Total Tele Modem	49.95
Mighty Mo	69.95
1660 Modem	49.95

INDUS

Indus GT for C-64	239

Animatidn Station

for C-64	[illegible]

PRINTER INTERFACES

Axiom	69.95
G-Wiz	59.95
Data Share PPI	49.95
Cardco G +	Call

C-64 SUPER SPECIALS

Consultant-D	**49.95**
Voice Messenger	**39.95**
Reach For The Stars-D	**29.95**
Star Rank Boxing-D	**20.95**
Computer Ambush-D	**37.95**
Carrier Force-D	**37.95**
Kamptgruppe-D	**37.95**
Carriers At War-D	**32.95**
Superbase 64-D	**44.95**

Super Special Prices Elleclive Now Through September 31, 1985

COMMODORE SOFTWARE

ACCESS

Beachhead-D	21.95
Beachhead II-D	24.95
Raid/Moscow-D	24.95
Mach V-Cart	21 95

BATTERIES INCLUDED

Consultant-D	64 95
Paperclip w/Spellpak-D	79 95
Super Busscard II	129.95
Home Drganizer Items-D	16 95
B.I. B0 Card	109 95
Cal-Kit	34.95
Home Pak-D	34 95

COMMODORE

Easy Calc-Cart	34 95
Easy Mail-D	19 95
Easy Script-D	44.95
Easy Spell-D	19 95
Logo-D	49 95
The Manager-D	34 95
Suspended-D	29 95
Starcross-D	29 95
Deadline-D	29 95
Cutthroats-D	29.95
The Suspect-D	29.95
Zork I, II or III-D	29 95

ELECTRDNIC ARTS

Sky Fox-D	24 95
Mail Drder Monster-D	24.95
Adv. Construction-D	29 95
Racing Destruction-D	24 95

INFDCOM

Enchanler-D	24.95
Planetfall-D	24.95
Sorcerer-D	29 95
Hitchiker-D	24.95
Wishbringer-D	29 95

SSI

Duestron-D	24 95
Phantasie-D	24 95
Germany 19B5-D	37 95
Wings of War-D	24 95
Fighter Command-D	37.95
Mech Brigade-D	37.95
Norway 1985-D	21.95

See Atari Section for Rest of Items and Prices

T-Cassette
D-Disk
Cart-Cartridge

MISC. COMMODORE

Flight Simulator II-D	34 95
Star Leag. Baseball-D/T	23.95
Mastertype-D/Cart	27 95
Strip Poker-D	23.95
Sam-D	41.95
Spelunker-D	20.95
Stealth-D	20.95
Bungling Bay-D	20.95
Dr. Creep-D	20 95
Dn-Field Tennis-D/T	23.95
F-15 Strike Eagle-D/T	23.95
Movie Maker-D	37 95
VIP Terminal-D	29.95
Doodle-D	27.95
Ghost Busters-D	24.95
Koala Printer-D	19.95
Karaleka-D	20.95
Print Shop-D	34.95
Amazon-D	22.95
Rendezvous/Rama-D	22.95
Farenhelt 451-D	22 95
Dragonworld-D	22.95
Shadow Keep-D	22.95
Spy vs Spy-D	23.95
Stress Relax Sys.-D	79.95
Blue Max-D/T	23.95
Ultima II-D	37.95
Ultima III-D	37.95
Ultima IV-D	41 95
Micro-League Baseball-D	29 95
Home Pak-D	34 95
Math Blaster-D	34 95
Word Attack-D	34 95
Speed Reader II	49 95
Stunt Flyer-D	Call
Sargon III-D	34 95
Super Zaxxon-D	23 95
Decision/Desert-D	27 95
Kennedy Approach-D	23 95
Crusade/Europe-D	27.95
Halley Projecl-D	29.95
Transylvannia-D	23.95
Bounty Bob-Cart	34.95
Blue Max 2001-D	23.95
Fleet System II-D	49.95
Evelyn Wood-D	34.95
Write Now-Cart	34.95
Bruce Lee-D	27.95
Conan-D	27.95
Fast Load-Cart	25.95
Net Worth-D	49 95

No surcharge for MasterCard or Visa

Classified

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.)

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

SOFTWARE

LOTTO PICKER. Go for Million Dollar Jackpots! Picks all USA Lotto games +! PRO FOOTBALL ANALYST. Beat the points consistently and easily! They pay for themselves! Each $35. IBM-PC/TI99 ver. RIDGE, 170 Broadway, #201-C, NYC, NY 10038. 718-833-6335. Free Catalog avail.

SPANISH GERMAN & FRENCH. 1000 drills. Extensive vocab/exciting games. Allows 600 user exercises. Apple II+/e/c 64K. Specify language. $24.95 + $1 Shipping. AET, P.O. Box 6591, Bridgewater, NJ 08807.

FIXED ASSET DEPRECIATION on Commodore 64. Handles acrs and pre-acrs methods. $39.95. For free info write: MPM Software Prods., P.O. Box 3522, Glendale, CA 91201.

FOOTBALL PREDICTION PROGRAM for C64 & IBM. Impress friends with picks calculated from newspaper stats, incl. turnovers, injuries, etc. Pkg. includes 1984 year-end Pro stats. Send $25.95 to: JZL Systems, 201 Brookside Ln, Somerville, NJ 08876

FREE EDUCATIONAL SOFTWARE CATALOG. 112 pp, color pictures, 1000+ programs for Com, Apple, IBM, Atari. Send name + address to: Interstate Software, P.O. Box 8952, Boise, ID 83707, (208) 342-3347

WORD WARS!—A crossword game for 1 or 2. Beats Monty a mile! Send for FREE how-to bklet. C64 cass. $24.95. No-hassle guar. Rollingwood, POB 364, Richmond, CA 94806

TI-99/4A Software/Hardware bargains. Hard to find items. Huge selection. Fast Service. Free Catalog. D.E.C., Box 690, Hicksville, NY 11801

SUM COINS C64 COIN EDU. SOFTWARE. GRADES. K-3. Number & type of coin options. Speed Drills, & number of problems. $12.95 cass. $14.95 disc. HIS KIDS SOFTWARE, 2895S. Mudd Creek Rd., Oak Harbor, Ohio 43449

WAR GAMES DIALER-ATARI 835/1030 MODEMS. Send $11.95 to MK Software
39 Eaton St., Buffalo, NY 14209
Personal checks take 2–4 wks. to clear.

ATARI USERS. 4 games on cassette, $25. Saucers + monsters, poker, seafinder, pinball. Write or call Keith Anderson, 8435 12th Ave., Silver Spring, MD 20903 (301) 434-0285

AMCUP, AN ORIGINAL sailboat racing game. 1 or 2 players. C-64 w/1541 + joystick. $14.95 ch/m.o. SASE for info. Dream Dog Software, P.O. Box 1724, Eugene, OR 97440.

"CRAPSMAN!" The C-64 casino craps tutor! Learn all bets. Test systems. Up to four players. $31.95 pp, disk. EMCEE SOFT CORP., P.O. Box 402124, Miami Beach, FL 33140.

SHOPPING LIST PROGRAM for C-64 with 1541 disk and printer. It prepares a shopping list for meals you select, organized by your market sections. Send $42 to 4955 Woodman Ave., Sherman Oaks, CA 91423

SOFTWARE PROTECTOR 64. Copy and list protect your valuable Basic programs. Only $10 (disk only). Systems Software, 44 Rte 25A #209, Smithtown, NY 11787.

PINOCHLE for Commodore 64, PC and PCjr. Real, double-deck, partnership pinochle gives you a partner and two opponents! On disk, $25 postpaid. Jim Bernard, 8018 Sunset Path Ct., Springfield, VA 22153.

TI CHECKWRITER Letter code and dollar amount are all that's necessary to print a complete check. Provides yearly totals for taxes. Disk $24.95. R.I.P. Software, 248 Purisima Rd., Woodside CA 94062.

SCIENCE SOFTWARE with graphics for the Commodore 64. Astronomy programs, etc. Catalog and demo disk $3. David Eagle, 7952 W. Quarto Dr., Littleton, CO 80123.

HARDWARE

Trade in your used Commodore or Atari on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure $1.50 and SASE. NEW WEST TECHNOLOGY, 4B Monroe Pkw., Box 200, Ste. 134, Lake Oswego, OR 97034

T.I. 99/4A: Full memory expansion for your T.I. Computer! New memory added entirely within the computer console! No plugs or external power needed. Write: Tim Northrup, 3305 Rutherglen, El Paso, Texas 79925

REPLACEMENT POWER SUPPLIES FOR C-64 or VIC-20. $29 ea. plus $3 shipping. 64 parts list $5-refundable w/1st order. AA Computer, 2726 Park St., JAX, FL 32205.

NUMERIC KEYPAD with joystick CURSOR control, gold contact $75. Centronics interface $40. Autodos cartridge automatically loads programs from easy-to-use menu $40. All C-64 fully compatible. SS 64 RF Mod. $40. Battery Back-up with best available protection for C-64 $50, Apple/IBM $150 (U.S.) APPLIED COMPUTING, Box 1566, PTBO, ONT. K9J 7H7. Send $2 for cat. Call (705) 745-8617 anytime.

MISCELLANEOUS

*YOU NEED A MAG-RAK! Ends awkwd fumbling when entering programs. Holds & lineguides an entire magazine! $34.95 V/Mstcd/Ck/MO—Stop suffering—Order yours today.
(213) 662-6905, R&L, Bx 26A89, LA, CA 90026

Maxell MD1, $1.39 - MDZ, $1.99. Dysan 104/1D, $1.79 - 104/2D, $2.39. Shipping $3.75. Also Verbatim, IBM, 3M, BASF. TAPE WORLD, 220 Spring St., Butler, PA 16001, 1-800-245-6000, Visa, MC.

SELL YOUR COMPUTER
National Listing Service: $19.95. Listed until sold. No commission fees. All types of computer equipment/peripherals.
1-800-TEC TRAN
Visa/MC accepted. In MA (617) 491-4888.

FREE! SHOP BY MODEM in our revolutionary electronic shopping mall. We even accept credit cards! We have products YOU want! Call 1-818-840-8066 with your modem NOW!

FLYING COMPUTERS but stuck on the ground? Learn requirements to fly aircraft, with or without license. $3.00 to: Aviation Resource, P.O. Box 369-C, Lancaster, NY 14086.

Super 99 Monthly—quality info for TI-99/4A & compatibles since 09/84. US $12/yr. Other $16/yr. Bytemaster Computer Svces., 171 Mustang, Sulphur LA 70663.

BUY-SELL-TRADE computers, hardware, software. Announce clubs, BBS's, etc. Send for free sample of Micro-Swap, the computer classifieds. P.O. Box 24, Esmond, IL 60129.

COMPUTER OWNERS!!! Earn dollars and save money with your computer. Write C&D Associates, Box 851, Mt. Prospect, IL 60056. Specify computer type. No Investment!!

TI-99/4A Software/Hardware bargains.
Hard to find items. Huge selection.
Fast Service. Free Catalog.
D.E.C., Box 690, Hicksville, NY 11801

HELP IS ON THE WAY!
Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

SAFEGUARD YOUR COMPUTER KEYBOARD from contaminants with OVERBYTE molded keyboard covers. $24.95 xparent & durable. Call (213) 866-2583/send $2 further info. OVERBYTE, P.O. Box 10652, Burbank, CA 91510.

PHONEMARK DATASETTES: C64, Plus4, Vic-20. $25.00 plus shipping. We are direct importers. American Mercantile Co., 2450 1st Ave., South, Seattle, WA 98134 (206) 624-6141.

DISK SERVICE MANUAL. Service floppies without special software or equipment. FREE information. Consumertronics-DSM, P.O. Drawer 537, Alamogordo, NM 88310.

$$ MONEY MAKER $$
Increase your income by hundreds and even thousands of dollars. Money back guarantee! (Apple, Macintosh, IBM) Free brochure: 1-800-223-5838 (305-771-6498).

PRODUCT MART

3M Diskettes
Lifetime Warranty

TIRED OF WAITING
FOR SERVICE AND PRICE?
9 out of 10 SURVEYED
DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

Formatted and hard sectored disks
~~in stock-Dealer inquiries invited.~~
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NH

**NORTH HILLS CORP.
INTERNATIONAL**
3564 Rolling View Dr.
White Bear Lake, MN. 55110
MN. call collect—612-770-0485

wabash DATATECH DISKETTES

Velue Priced Diskettes!

Lifetime Warranty! Hub Rings! 100% Error-Free!

5¼" Diskettes—Boxed 10/Box

SS SD	$ .97 Each
SS DD	$1.07 Each
DS DD	$1.17 Each
HD for PC/AT	$2.40 Each

3½" Diskettes

SS	$2.25 Each
DS	$2.95 Each

SUPER-SAVINGS ON POLY-PACK DISKETTES
With Envelopes, WP Tabs, User ID Labels

SS SD	$.69 Each
SS DD	$.79 Each
DS DD	$.89 Each

Minimum order — 50 Diskettes Quantity Discounts Available. MI Residents, add 4% Sales Tax. Shipping & Handling $4 00/100 Diskettes. C O D., add $3.00 Cert Check/Money Order

TO ORDER: Call or Write…

pd™

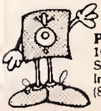
Precision Data Products
P.O. Box 8387
Grand Rapids, MI 49508
(616) 452-3457
Michigan 1-800-632-2468
Outside Mich. 1-800-258-0028

VISA **C.O.D.** MasterCard

Maxell Floppy Disks

The Mini-Disks
with maximum quality.

Dealer inquiries invited. C.O.D's accepted
Call FREE (800) 235-4137



PACIFIC EXCHANGES
100 Foothill Blvd., San Luis
San Luis Obispo, CA 93401
In Cal. call (800) 592-5935 or
(805)543-1037

DISKS 75¢

QUALITY MEDIA	Because we buy
LIFETIME REPLACE	in huge volume to
GUARANTEE	supply software
HUB RINGS, TYV. EPS.	manufacturers, our
WRITE PROTECTS	prices can't be beat.

		50+	250+	500+	1000+
5.25" SSDD	1.00	90	.85	.80	.75
5.25" DSDD	1 15	1.05	1.00	.95	.85
PC FORMATTED	1.45	1.35	1.25	1.15	1.10
3 5" 1D (For Mac)	2.70	2.80	2 45	CALL	CALL
3.5" 2D (For HP)	3 50	3.25	3.15	CALL	CALL

BLACKSHIP
COMPUTER SUPPLY

P.O Box 883362 • San Francisco, CA 94188
Inquiries + No. Cal. orders 415-550-0512
So. Cal. orders 800-235-6647 x31
USA orders 800-235-6646 x 31

Add $3.00 shipping and handling
(CA residents add 6.5% sales tax) VISA/MC/COD

VOICE COMMAND YOUR MICRO

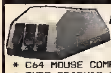
NEW!

SMART EARS II

- SMART EARS II IS A FAST
- STATE OF THE ART VOICE
- RECOGNITION MODULE. ITS THE
- WORLDS FIRST VOICE COMMAND
- JOYSTICK;MANY GAMES CAN BE PLAYED
- WITH YOUR VOICE,IMAGINE THAT!
- CAN BE USED TO CONTROL BSR
- MODULES,TELEPHONE DIALERS &
- ROBOTS.COMBINED WITH OPTIONAL MODULES
- YOUR COMPUTER COMES AL1VE!
- INCLUDES SOFTWARE AND HEADSET MIC.

LEFT! STOP! RIGHT!

$69



MAGIC 64 MOUSE
$69

- C64 MOUSE COMPLETE WITH ICON
- TYPE GRAPHIC SOFTWARE AND
- SPRITE DESIGNER , 16 COLORS

FREE

FREE ROBOT WITH YOUR ORDER

ADD $4 SHIPPING/C.A. RES ADD 6.5% TAX

TVTRON
4959 SUNSET BLVD
LOS ANGELES C.A.90027

TELEPHONE
(213) 6607811

maxell DISKS
LIFETIME WARRANTY

TIRED OF WAITING
FOR SERVICE AND PRICE?
9 out of 10 SURVEYED
DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

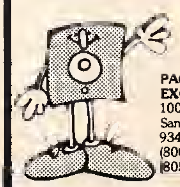
Formatted and hard sectored disks
in stock-Dealer inquiries invited.
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NH

**NORTH HILLS CORP.
INTERNATIONAL**
3564 Rolling View Dr.
White Bear Lake, MN. 55110
MN. call collect—612-770-0485

Verbatim flexible disks

Call Free **(800) 235-4137** for prices and information. Dealer inquiries invited. C.O.D. and charge cards accepted.



VISA

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. In Cal. call
(800) 592-5935 or
(805) 543-1037.

CAPUTE!

Modifications or Corrections To Previous Articles

Animator For Apple And IBM

In the August issue, eleven program lines were inadvertently omitted from the Apple version of this graphics utility (BASIC portion, Program 6, p. 58). The missing lines are as follows:

```
Ø6 1Ø3Ø E%(J,I) = O: FOR Q = Ø T
        O 6:T = INT (O / 2): PRI
        NT CHR$ (46 + 13 * (O -
        T * 2));:O = T: NEXT : N
        EXT : IF I < 23 THEN PRI
        NT
38 1Ø4Ø NEXT : RETURN
C7 1Ø5Ø POKE 242,Ø: CALL 32777,Ø
        : GOSUB 1Ø1Ø: CALL 32768
        ,Ø,2Ø6,12: RETURN
DC 1Ø6Ø CALL 32768,A,2Ø6,12
28 1Ø7Ø VTAB 1: HTAB 27: PRINT "
        ONE MOMENT";
DF 1Ø8Ø CALL 32774,A: FOR I = Ø
        TO 23: FOR J = Ø TO 2: I
        NPUT "";E%(J,I): NEXT :
        NEXT : CALL 327BØ
7E 1Ø9Ø HOME : FOR I = Ø TO 23:
        FOR J = Ø TO 2:O = E%(J,
        I)
92 11ØØ FOR Q = Ø TO 6:T = INT (
        O / 2): PRINT CHR$ (46 +
         13 * (O - T * 2));:O =
        T: NEXT : NEXT : IF I <
        23 THEN PRINT
DC 111Ø NEXT : HTAB 27: VTAB 1:
        PRINT SPC( 1Ø): RETURN
86 112Ø GOSUB 56Ø: GOSUB 7ØØ: VT
        AB 19: HTAB 1Ø: PRINT "I
        NSERT BOX ";A;: GOSUB 11
        6Ø: IF C = 2Ø6 THEN 115Ø
```

The last line of the IBM version (Program 1, p. 52) was partially obscured. It should read as follows:

```
CM 25Ø4Ø A$=INKEY$:IF A$<>" " TH
     EN 25Ø4Ø ELSE RETURN
```

Atari List Scroller

This utility program in the July issue (p. 68) will crash because of a line numbering problem. Line 32702 should be revised as follows:

```
327Ø2 LNUM=PEEK(A)+PEEK(A
      +1)*256:IF LNUM>=32
      7ØØ THEN 327Ø4
```

Thanks to William Webb and others who pointed this out.

IBM Proofreader

A bug was uncovered in our IBM "Automatic Proofreader," published in "COMPUTE!'s Guide to Typing In Programs" since October 1984. It has been hidden until now because it appears only when the first characters following the line number in a program line are either D or E followed by a number, as is the case in lines 110 and 120 of Program 3 from "Viewports in IBM BASIC" (July issue, p. 71). In these cases, the VAL function in line 190 interprets the characters as indicating exponential notation, leading to an incorrect line number. The solution, suggested by reader Daniel Norling, is to make the following additions and changes to the Proofreader:

```
AG 19Ø REM
JB 2Ø5 BL=INSTR(L$," "):IF BL=Ø
       THEN BL$=L$:GOTO 2Ø6 ELSE
        BL$=LEFT$(L$,BL-1)
GH 2Ø6 LNUM=VAL(BL$):TEXT$=MID$(
       L$,LEN(STR$(LNUM))+1)
KA 47Ø WHILE NOT EOF(1):LINE INP
       UT #1,L$:BL=INSTR(L$," ")
       :BL$=LEFT$(L$,BL-1):LNUM(
       P)=VAL(BL$):L$(P)=MID$(L$
       ,LEN(STR$(VAL(BL$)))+1):P
       =P+1:WEND
```

Apple Universal INPUT

There is an error in the machine language for this INPUT enhancement routine from the June issue (p. 91), although you can use the routine with no problems most of the time. As reader Don Andrews discovered, the bug becomes apparent only when you attempt to input a string more than 76 characters long. (An LDY $00 instruction was used where an LDY #$00 was required.) The routine can be fixed by changing the 164 in line 280 to a 160:

```
28Ø DATA 3Ø,3,16Ø,Ø,2Ø4,3
     Ø,3,24Ø
```

A review of *HomePak* in the July issue mentioned a free upgrade for those who bought the first version. (The upgraded telecommunications portion of the program now dials most Commodore modems.) However, the upgrade does require a $10 shipping and handling fee and the return of the original disk. Write to Batteries Included at 30 Mural Street, Richmond Hill, Ontario, L4B 1B5, Canada, or 17875 Sky Park North, Suite P, Irvine, CA 92714. ©

Advertisers Index

For COMMODORE and ATARI

WHEN BATTERIES INCLUDED SET OUT TO DESIGN THE VERY BEST WORD PROCESSOR FOR ATARI® COMPUTERS... THEY FOUND THEY ALREADY HAD IT.

PaperClip
Professional Word-Processor
BATTERIES INCLUDED
"The Energized Software Company!"
for ATARI
PaperClip Professional Word-Processor™
PAPERCLIP
Features:
Dual text window editing
On-screen help files for all key program functions
Complete with printer files for all major printers
Automatically generates a Table of Contents
Auto save capability
And much, much more...
ATARI 48K DISK

Features all standard word processing functions PLUS

- Full Screen Editing
- Dual text windows with cut and paste between windows.
- Video preview with scrolling, lets you see exactly what will be printed, *before* it is printed
- Transfer text between documents – easily, simply
- Horizontal scrolling – create wide documents with ease
- Re-arrange words, phrases, sentences, paragraphs, and columns with just a few keystrokes
- More than 40 different printer drivers – plus printer editor allows programming for any conceivable new printer
- Easy to learn, easy to use

COMPATIBLE WITH ATARI 400, 800, 600 XL, 800 XL, 1200 XL, 65 XE, 130 XE

PAPERCLIP™

Few word processors have allowed Atari users to tap the full resources of their computer ***until Atari Paper Clip***... *Atari Paper Clip is an extremely powerful, fully featured word processor that will allow your Atari to operate to the limits of its potential, with an ease of operation and speed you've never thought possible.*

PAPERCLIP FOR ATARI® AND COMMODORE® OWNERS WHO WANT THE VERY BEST IN WORD PROCESSING.

BATTERIES INCLUDED

30 Mural Street
Richmond Hill, Ontario
L48 1B5 CANAOA
(416) 881-9941
Telex: 06-21-8290

"The Energized Software Company!"

17875 Sky Park North, Suite P
Irving, California
USA 92714
(416) 881-9816
Telex: 509-139

WRITE TO US FOR FULL COLOUR CATALOGUE of our products for COMMODORE, ATARI, APPLE and IBM SYSTEMS

FOR TECHNICAL SUPPORT OR PROOUCT INFORMATION PLEASE PHONE (416) 881-9816

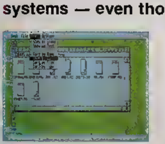
AMERICA WE BUILT IT FOR YOU

THE ATARI 520ST

"We promised.
We delivered.
With pride,
determination,
and good old
ATARI know how."
*Sam Tramiel,
President,
ATARI CORP.*

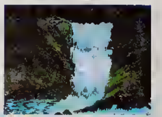
No other computer we know of has been awaited with such anticipation, has received so much national and trade press, and has been so unanimously acclaimed — as the remarkable 520ST.

And for good reason. Its development represents a bold, new standard in personal computing power. The 520ST simply obsoletes all current personal systems — even those costing thousands of dollars more. Beneath its full stroke 94-key keyboard is an operating environment so intelligent that it puts you in command almost at once. It's that easy.



Graphic symbols quickly identify available functions. Menus appear just by aiming the mouse. Point to a specific operation, click the mouse and instantly you are able to develop full color charts, recall files from within folders, and so much more.

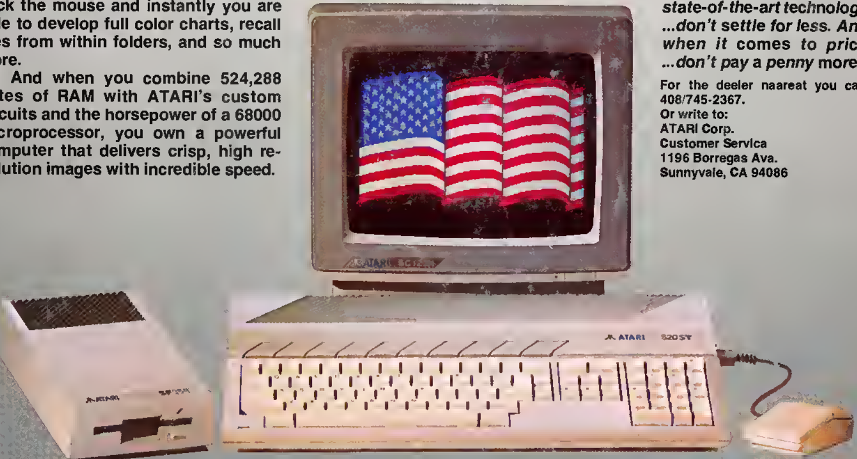
And when you combine 524,288 bytes of RAM with ATARI's custom circuits and the horsepower of a 68000 microprocessor, you own a powerful computer that delivers crisp, high resolution images with incredible speed.



With a monochrome monitor your 520ST displays 640 x 400 pixels of extremely high resolution clarity. Select a color monitor and you are able to create beautiful graphs and diagrams from a palette of 512 vivid colors.

Power to grow. An array of expansion ports allow you to easily customize your 520ST. There are standard serial and parallel interface ports for connecting printers and telecommunications equipment, MIDI connectors to interface with music synthesizers and keyboards, and 3.5 inch floppy disk, cartridge and joystick ports. There is also a hard disk port with the lightening communications speed of 1.33 Megabytes per second. ATARI 520ST systems are available now. *When it comes to state-of-the-art technology ...don't settle for less. And when it comes to price ...don't pay a penny more.*

For the deeler naareat you call 408/745-2367.
Or write to:
ATARI Corp.
Customer Servica
1196 Borregas Ava.
Sunnyvale, CA 94086



The New ATARI®

Power Without the Price

\$799^{95}

*plus state and local taxes where applicable.

SYSTEM INCLUDES: 520ST Personal Computer, Monochrome Monitor, Mouse Controller, 3.5 inch Disk Drive, TOS™ - The Operating System Disk
ATARI Logo™ Language Disk
With full color monitor: $999.95*